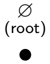
Tree diagrams: A decision tree consists of a "root", a number of "branches" leaving the root, and possible additional branches leaving the endpoints of other branches (usually drawn upside-down). We use a branch to represent each possible choice. We represent the possible outcomes by "leaves", the endpoints of branches not having other branches starting at them. (See §6.1)

Tree diagrams: A decision tree consists of a "root", a number of "branches" leaving the root, and possible additional branches leaving the endpoints of other branches (usually drawn upside-down). We use a branch to represent each possible choice. We represent the possible outcomes by "leaves", the endpoints of branches not having other branches starting at them. (See §6.1)

Example: How many strings of length-three of 1's and 0's do not have two consecutive 1's?

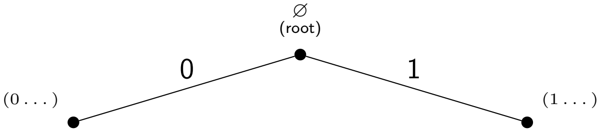
Tree diagrams: A decision tree consists of a "root", a number of "branches" leaving the root, and possible additional branches leaving the endpoints of other branches (usually drawn upside-down). We use a branch to represent each possible choice. We represent the possible outcomes by "leaves", the endpoints of branches not having other branches starting at them. (See §6.1)

Example: How many strings of length-three of 1's and 0's do not have two consecutive 1's?



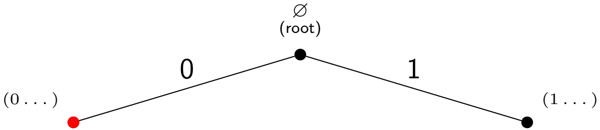
Tree diagrams: A decision tree consists of a "root", a number of "branches" leaving the root, and possible additional branches leaving the endpoints of other branches (usually drawn upside-down). We use a branch to represent each possible choice. We represent the possible outcomes by "leaves", the endpoints of branches not having other branches starting at them. (See §6.1)

Example: How many strings of length-three of 1's and 0's do not have two consecutive 1's?



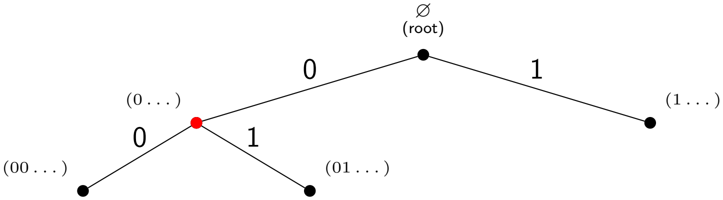
Tree diagrams: A decision tree consists of a "root", a number of "branches" leaving the root, and possible additional branches leaving the endpoints of other branches (usually drawn upside-down). We use a branch to represent each possible choice. We represent the possible outcomes by "leaves", the endpoints of branches not having other branches starting at them. (See §6.1)

Example: How many strings of length-three of 1's and 0's do not have two consecutive 1's?



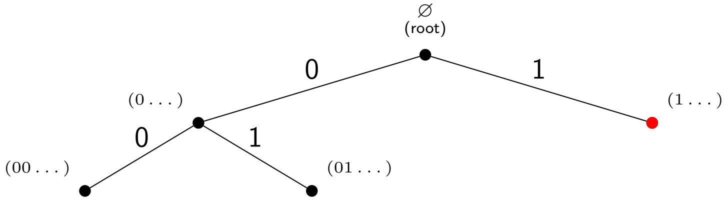
Tree diagrams: A decision tree consists of a "root", a number of "branches" leaving the root, and possible additional branches leaving the endpoints of other branches (usually drawn upside-down). We use a branch to represent each possible choice. We represent the possible outcomes by "leaves", the endpoints of branches not having other branches starting at them. (See §6.1)

Example: How many strings of length-three of 1's and 0's do not have two consecutive 1's?



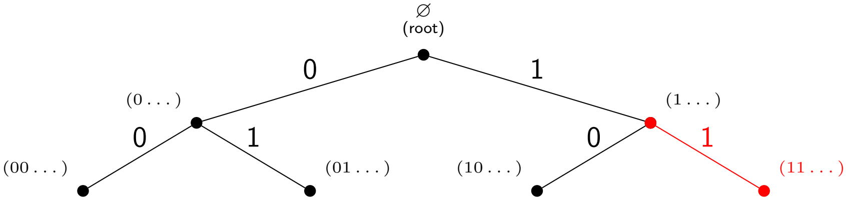
Tree diagrams: A decision tree consists of a "root", a number of "branches" leaving the root, and possible additional branches leaving the endpoints of other branches (usually drawn upside-down). We use a branch to represent each possible choice. We represent the possible outcomes by "leaves", the endpoints of branches not having other branches starting at them. (See §6.1)

Example: How many strings of length-three of 1's and 0's do not have two consecutive 1's?



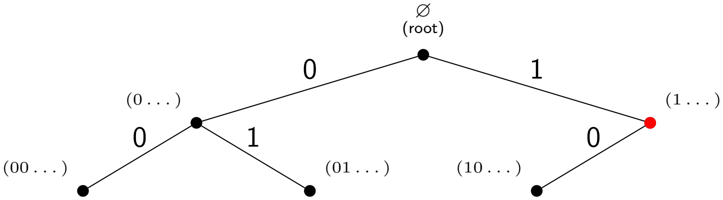
Tree diagrams: A decision tree consists of a "root", a number of "branches" leaving the root, and possible additional branches leaving the endpoints of other branches (usually drawn upside-down). We use a branch to represent each possible choice. We represent the possible outcomes by "leaves", the endpoints of branches not having other branches starting at them. (See §6.1)

Example: How many strings of length-three of 1's and 0's do not have two consecutive 1's?



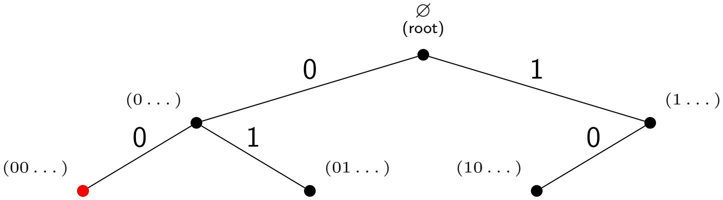
Tree diagrams: A decision tree consists of a "root", a number of "branches" leaving the root, and possible additional branches leaving the endpoints of other branches (usually drawn upside-down). We use a branch to represent each possible choice. We represent the possible outcomes by "leaves", the endpoints of branches not having other branches starting at them. (See §6.1)

Example: How many strings of length-three of 1's and 0's do not have two consecutive 1's?



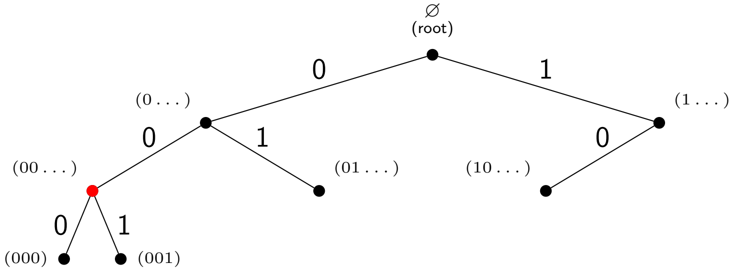
Tree diagrams: A decision tree consists of a "root", a number of "branches" leaving the root, and possible additional branches leaving the endpoints of other branches (usually drawn upside-down). We use a branch to represent each possible choice. We represent the possible outcomes by "leaves", the endpoints of branches not having other branches starting at them. (See §6.1)

Example: How many strings of length-three of 1's and 0's do not have two consecutive 1's?



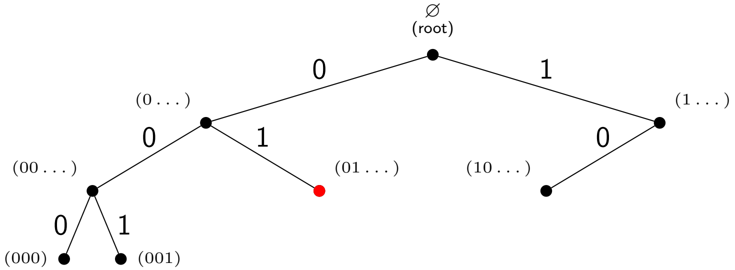
Tree diagrams: A decision tree consists of a "root", a number of "branches" leaving the root, and possible additional branches leaving the endpoints of other branches (usually drawn upside-down). We use a branch to represent each possible choice. We represent the possible outcomes by "leaves", the endpoints of branches not having other branches starting at them. (See §6.1)

Example: How many strings of length-three of 1's and 0's do not have two consecutive 1's?



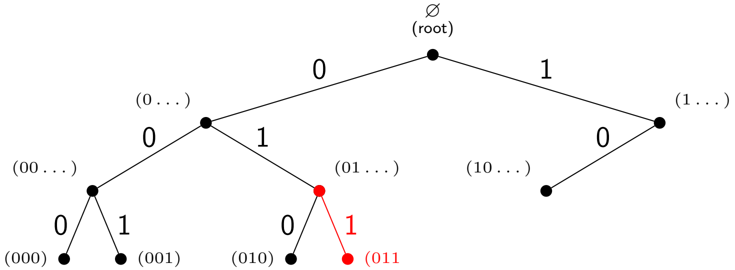
Tree diagrams: A decision tree consists of a "root", a number of "branches" leaving the root, and possible additional branches leaving the endpoints of other branches (usually drawn upside-down). We use a branch to represent each possible choice. We represent the possible outcomes by "leaves", the endpoints of branches not having other branches starting at them. (See §6.1)

Example: How many strings of length-three of 1's and 0's do not have two consecutive 1's?



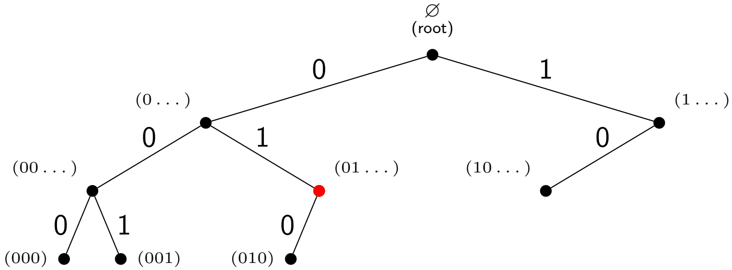
Tree diagrams: A decision tree consists of a "root", a number of "branches" leaving the root, and possible additional branches leaving the endpoints of other branches (usually drawn upside-down). We use a branch to represent each possible choice. We represent the possible outcomes by "leaves", the endpoints of branches not having other branches starting at them. (See §6.1)

Example: How many strings of length-three of 1's and 0's do not have two consecutive 1's?



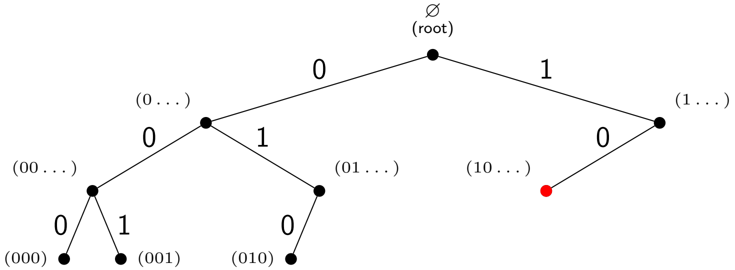
Tree diagrams: A decision tree consists of a "root", a number of "branches" leaving the root, and possible additional branches leaving the endpoints of other branches (usually drawn upside-down). We use a branch to represent each possible choice. We represent the possible outcomes by "leaves", the endpoints of branches not having other branches starting at them. (See §6.1)

Example: How many strings of length-three of 1's and 0's do not have two consecutive 1's?



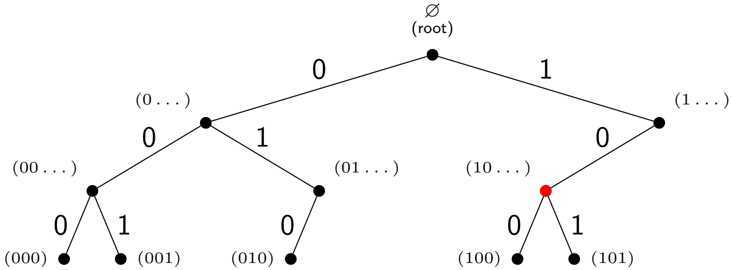
Tree diagrams: A decision tree consists of a "root", a number of "branches" leaving the root, and possible additional branches leaving the endpoints of other branches (usually drawn upside-down). We use a branch to represent each possible choice. We represent the possible outcomes by "leaves", the endpoints of branches not having other branches starting at them. (See §6.1)

Example: How many strings of length-three of 1's and 0's do not have two consecutive 1's?



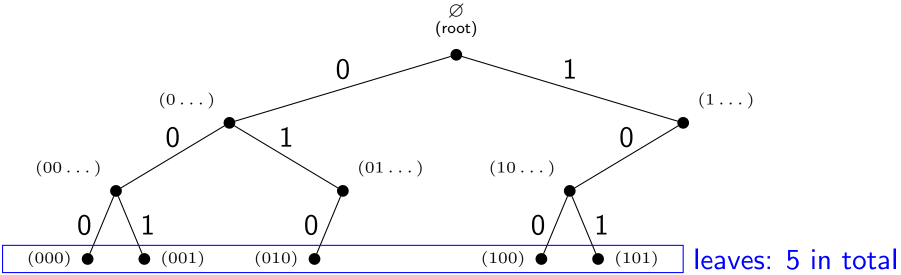
Tree diagrams: A decision tree consists of a "root", a number of "branches" leaving the root, and possible additional branches leaving the endpoints of other branches (usually drawn upside-down). We use a branch to represent each possible choice. We represent the possible outcomes by "leaves", the endpoints of branches not having other branches starting at them. (See §6.1)

Example: How many strings of length-three of 1's and 0's do not have two consecutive 1's?

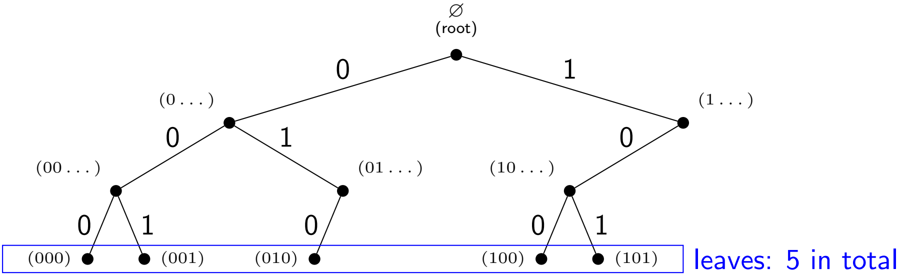


Tree diagrams: A decision tree consists of a "root", a number of "branches" leaving the root, and possible additional branches leaving the endpoints of other branches (usually drawn upside-down). We use a branch to represent each possible choice. We represent the possible outcomes by "leaves", the endpoints of branches not having other branches starting at them. (See §6.1)

Example: How many strings of length-three of 1's and 0's do not have two consecutive 1's?



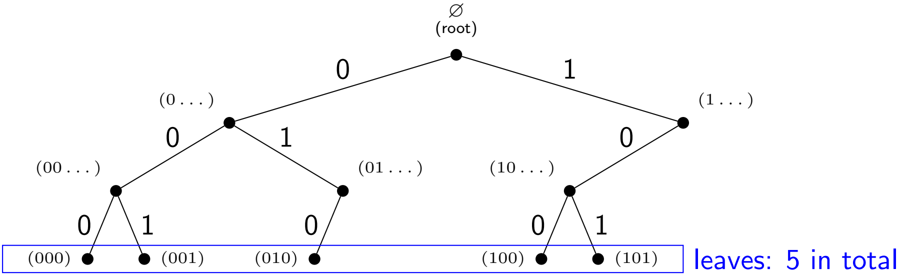
Tree diagrams: A decision tree consists of a "root", a number of "branches" leaving the root, and possible additional branches leaving the endpoints of other branches (usually drawn upside-down). We use a branch to represent each possible choice. We represent the possible outcomes by "leaves", the endpoints of branches not having other branches starting at them. (See §6.1)
Example: How many strings of length-three of 1's and 0's do not have two consecutive 1's?



See also Examples 21-23 in section 6.1. Note: the book labels the nodes by the choice; we label the edges by the choice and the nodes by the outcomes.

Tree diagrams: A decision tree consists of a "root", a number of "branches" leaving the root, and possible additional branches leaving the endpoints of other branches (usually drawn upside-down). We use a branch to represent each possible choice. We represent the possible outcomes by "leaves", the endpoints of branches not having other branches starting at them. (See §6.1)

Example: How many strings of length-three of 1's and 0's do not have two consecutive 1's?



See also Examples 21-23 in section 6.1. Note: the book labels the nodes by the choice; we label the edges by the choice and the nodes by the outcomes.

You try: Exercise 29

Applications of recurrence relations

Recall that a recursive definition for a sequence is an expression of a_n using the previous terms:

For example: $\underbrace{a_n = 3a_{n-1} + a_{n-3} + 1}_{\text{recurrence relation}}$ $\underbrace{a_0 = 1,\ a_1 = 15,\ a_2 = 0}_{\text{initial conditions}}$

Applications of recurrence relations

Recall that a recursive definition for a sequence is an expression of a_n using the previous terms:

For example: $\underbrace{a_n = 3a_{n-1} + a_{n-3} + 1}_{\text{recurrence relation}}$ $\underbrace{a_0 = 1,\ a_1 = 15,\ a_2 = 0}_{\text{initial conditions}}$

Example (Fibonacci's Rabbits)

Put two rabbits on an island. A pair of rabbits won't breed until they're 2 months old. Each mature pair of rabbits will produce a new pair of rabbits the following month. How many pairs of rabbits are there after n months? (assume balanced sexes)

Applications of recurrence relations

Recall that a recursive definition for a sequence is an expression of a_n using the previous terms:

For example: $\underbrace{a_n = 3a_{n-1} + a_{n-3} + 1}_{\text{recurrence relation}}$ $\underbrace{a_0 = 1,\ a_1 = 15,\ a_2 = 0}_{\text{initial conditions}}$

Example (Fibonacci's Rabbits)

Put two rabbits on an island. A pair of rabbits won't breed until they're 2 months old. Each mature pair of rabbits will produce a new pair of rabbits the following month. How many pairs of rabbits are there after n months? (assume balanced sexes)

Recurrence relation:

Number of pairs at month n	$=$	The number of pairs already around from month $n-1$	$+$	The number of eligible parenting pairs at month n (rabbits around since month $n-2$)

Applications of recurrence relations

Example (Fibonacci's Rabbits)

Put two rabbits on an island. A pair of rabbits won't breed until they're 2 months old. Each mature pair of rabbits will produce a new pair of rabbits the following month. How many pairs of rabbits are there after n months? (assume balanced sexes)

Recurrence relation:

Number of pairs at month n	$=$	The number of pairs already around from month $n-1$	$+$	The number of eligible parenting pairs at month n (rabbits around since month $n-2$)
a_n	$=$	a_{n-1}	$+$	a_{n-2}

Applications of recurrence relations

Example (Fibonacci's Rabbits)

Put two rabbits on an island. A pair of rabbits won't breed until they're 2 months old. Each mature pair of rabbits will produce a new pair of rabbits the following month. How many pairs of rabbits are there after n months? (assume balanced sexes)

Recurrence relation:

Number of pairs at month n	$=$	The number of pairs already around from month $n-1$	$+$	The number of eligible parenting pairs at month n (rabbits around since month $n-2$)
a_n	$=$	a_{n-1}	$+$	a_{n-2}

Initial conditions: Start with 1 pair. Still have 1 pair in the 1st month. Then the 1 pair starts to breed.

Applications of recurrence relations

Example (Fibonacci's Rabbits)

Put two rabbits on an island. A pair of rabbits won't breed until they're 2 months old. Each mature pair of rabbits will produce a new pair of rabbits the following month. How many pairs of rabbits are there after n months? (assume balanced sexes)

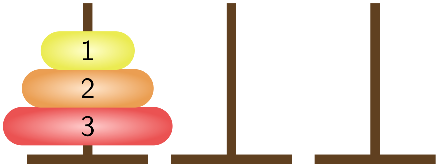
Recurrence relation:

Number of pairs at month n	=	The number of pairs already around from month $n-1$	+	The number of eligible parenting pairs at month n (rabbits around since month $n-2$)
a_n	$=$	a_{n-1}	$+$	a_{n-2}

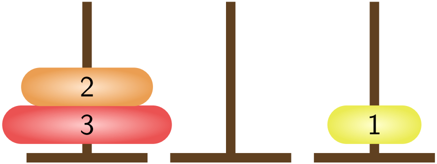
Initial conditions: Start with 1 pair. Still have 1 pair in the 1st month. Then the 1 pair starts to breed.

$$a_0 = 1, \qquad a_1 = 1$$

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.

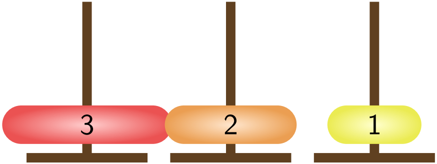


Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



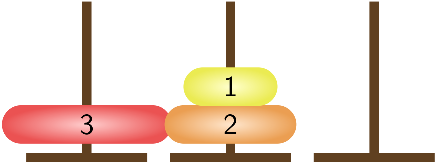
Moved disc from pole 1 to pole 3.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



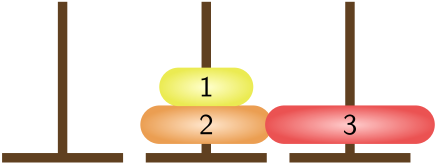
Moved disc from pole 1 to pole 2.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



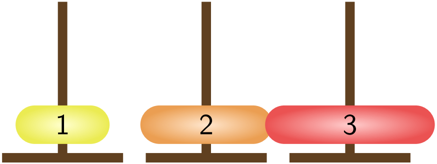
Moved disc from pole 3 to pole 2.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



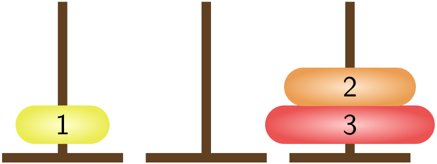
Moved disc from pole 1 to pole 3.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



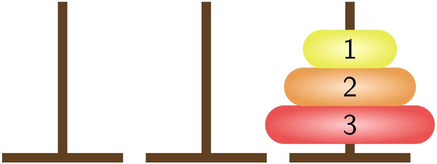
Moved disc from pole 2 to pole 1.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 2 to pole 3.

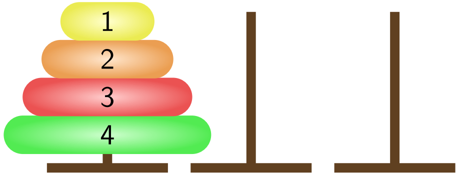
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



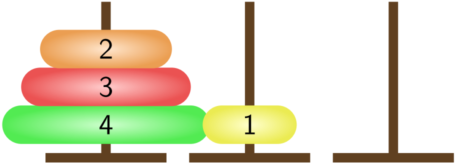
Moved disc from pole 1 to pole 3.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.

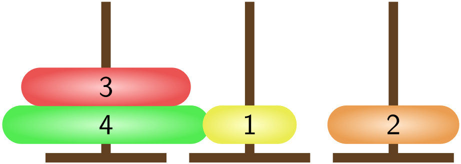


Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



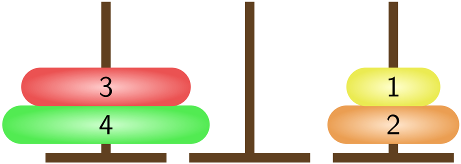
Moved disc from pole 1 to pole 2.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



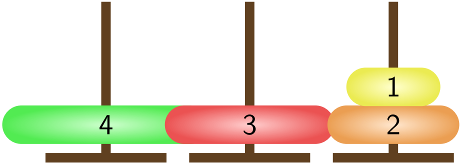
Moved disc from pole 1 to pole 3.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



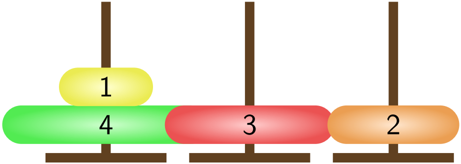
Moved disc from pole 2 to pole 3.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



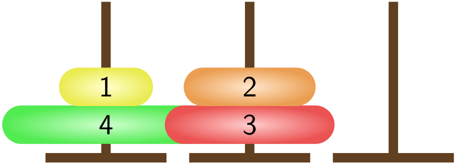
Moved disc from pole 1 to pole 2.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



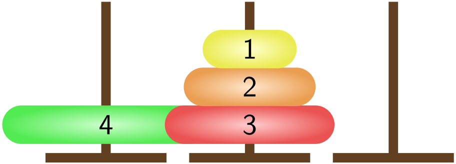
Moved disc from pole 3 to pole 1.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



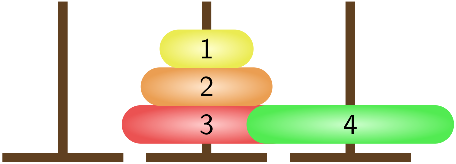
Moved disc from pole 3 to pole 2.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



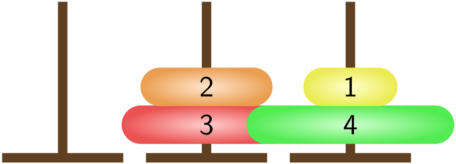
Moved disc from pole 1 to pole 2.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



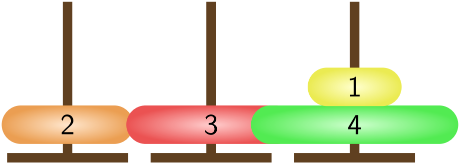
Moved disc from pole 1 to pole 3.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



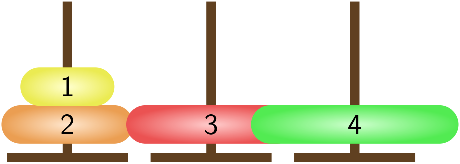
Moved disc from pole 2 to pole 3.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



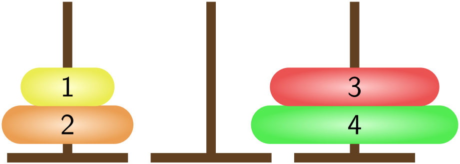
Moved disc from pole 2 to pole 1.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



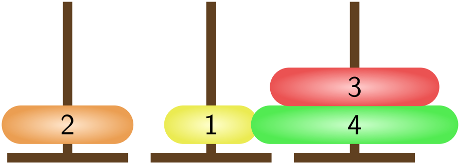
Moved disc from pole 3 to pole 1.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



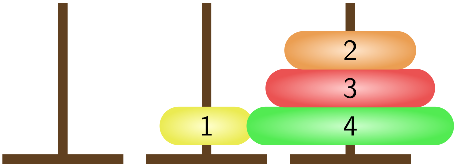
Moved disc from pole 2 to pole 3.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



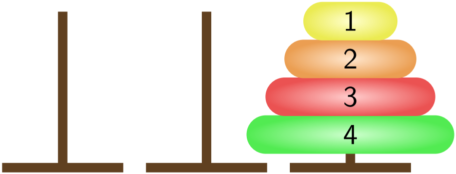
Moved disc from pole 1 to pole 2.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 1 to pole 3.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



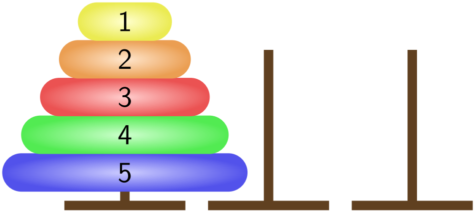
Moved disc from pole 2 to pole 3.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.

Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.

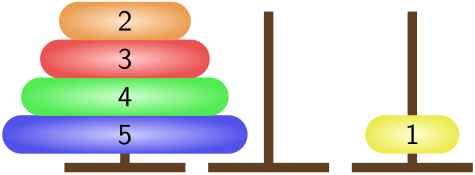
Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

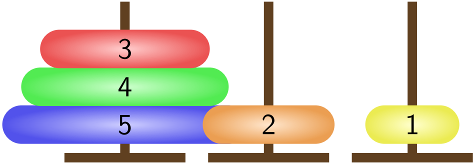
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 1 to pole 3.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

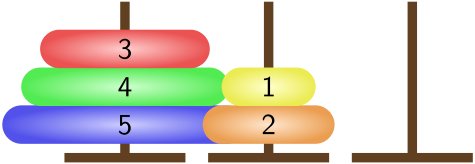
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 1 to pole 2.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

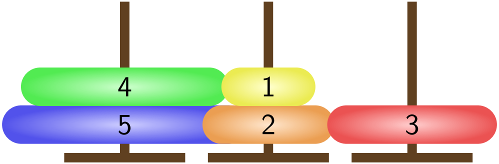
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 3 to pole 2.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

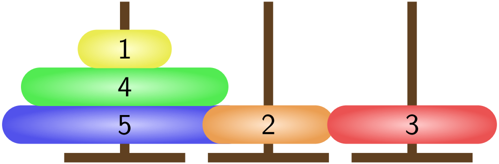
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 1 to pole 3.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

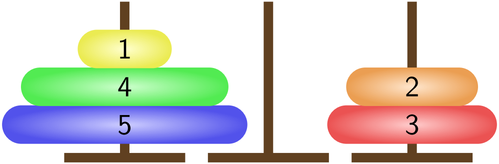
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 2 to pole 1.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

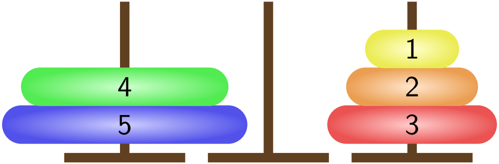
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 2 to pole 3.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

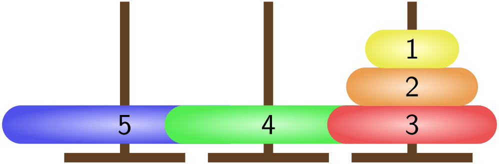
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 1 to pole 3.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

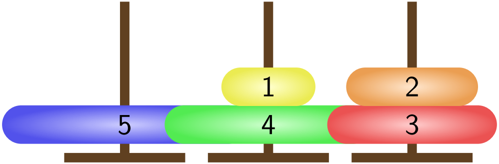
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 1 to pole 2.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

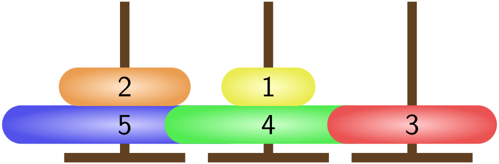
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 3 to pole 2.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

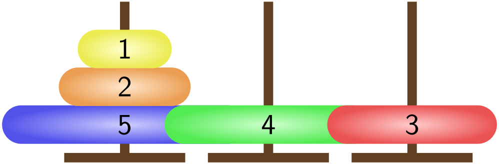
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 3 to pole 1.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

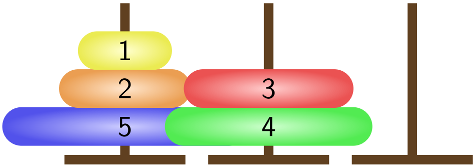
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 2 to pole 1.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

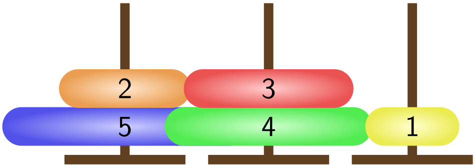
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 3 to pole 2.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

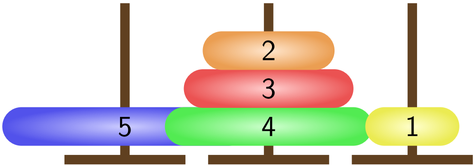
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 1 to pole 3.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

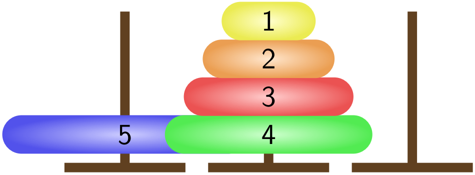
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 1 to pole 2.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

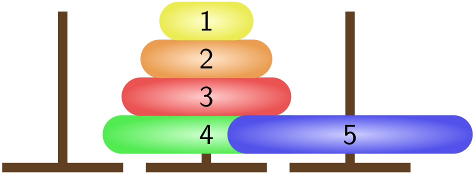
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 3 to pole 2.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

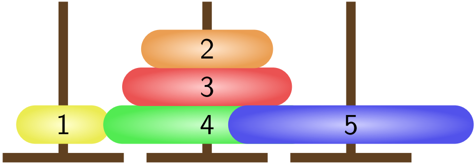
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 1 to pole 3.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

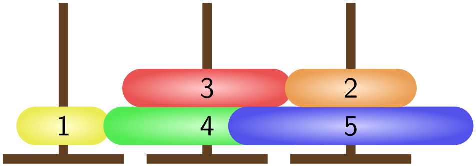
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 2 to pole 1.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n - 1$ discs on top.

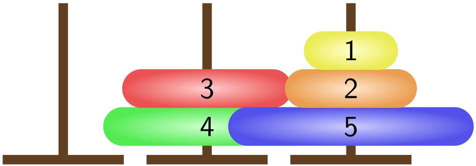
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 2 to pole 3.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

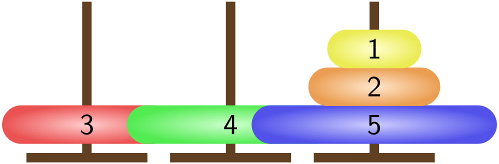
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 1 to pole 3.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

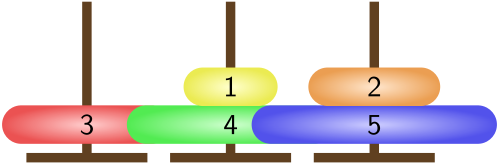
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 2 to pole 1.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

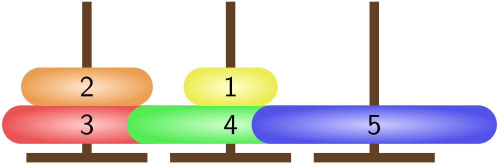
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 3 to pole 2.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

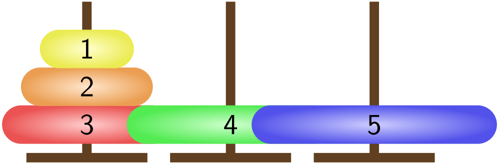
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 3 to pole 1.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

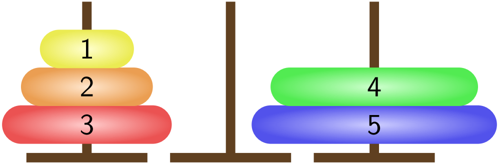
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 2 to pole 1.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

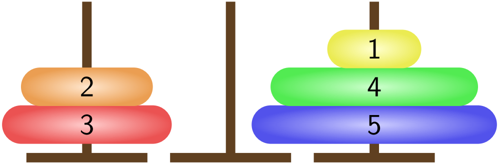
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 2 to pole 3.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n - 1$ discs on top.

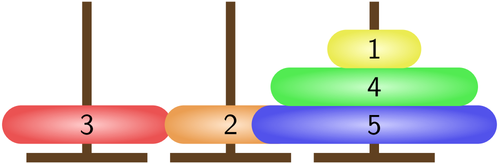
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 1 to pole 3.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

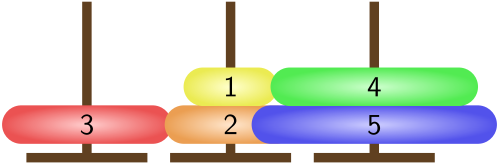
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 1 to pole 2.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

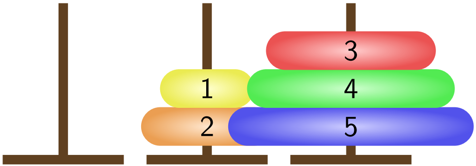
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 3 to pole 2.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

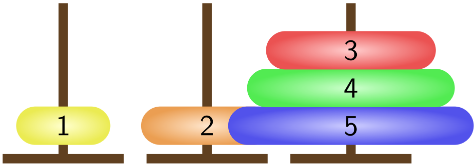
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 1 to pole 3.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n - 1$ discs on top.

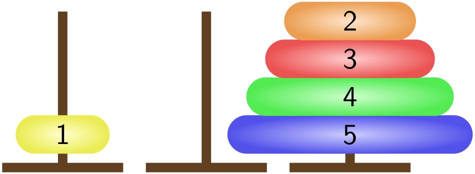
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 2 to pole 1.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n - 1$ discs on top.

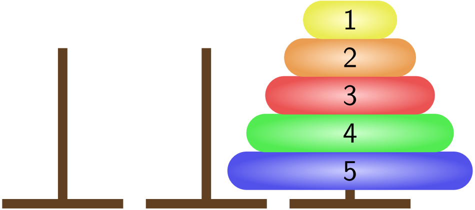
Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 2 to pole 3.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.



Moved disc from pole 1 to pole 3.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

Let H_n be the be the number of moves needed to solve the puzzle.

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

Let H_n be the be the number of moves needed to solve the puzzle.

Recursion relation: $H_n = H_{n-1} + H_{n-1} + 1 = 2H_{n-1} + 1$

Towers of Hanoi: Start with n discs of different sizes on the left-most of three pegs, in increasing order of size top-to-bottom. Move discs from pole to pole one at a time. End with all n discs on the right-most of three pegs, again in decreasing order top-to-bottom.

Solution: We solved these by first piling the all but one of the discs onto pole 2 (which takes the same number of moves are piling all but one onto pole 3); then we moved the biggest disc; then we moved the $n-1$ discs on top.

Let H_n be the be the number of moves needed to solve the puzzle.

Recursion relation: $H_n = H_{n-1} + H_{n-1} + 1 = 2H_{n-1} + 1$

Initial condition: $H_1 = 1$

Bit strings

Find a recurrence relation and give initial conditions for the number of bit strings of length n that do not have two consecutive 0s. How many such bit strings are there of length five?

Bit strings

Find a recurrence relation and give initial conditions for the number of bit strings of length n that do not have two consecutive 0s. How many such bit strings are there of length five?

Examples:

$n = 1: \ \{0, 1\}$

$n = 2: \ \{01, 10, 11\}$

$n = 3: \ \{010, 011, 101, 110, 111\}$

Bit strings

Find a recurrence relation and give initial conditions for the number of bit strings of length n that do not have two consecutive 0s. How many such bit strings are there of length five?

Examples:

$n = 1:\ \{0, 1\}$

$n = 2:\ \{01, 10, 11\}$

$n = 3:\ \{010, 011, 101, 110, 111\}$

Solution: For $n \geqslant 3$, break into cases, whether an admissible string ends in a 1 or a 0.

Bit strings

Find a recurrence relation and give initial conditions for the number of bit strings of length n that do not have two consecutive 0s. How many such bit strings are there of length five?

Examples:

$n = 1: \ \{0, 1\}$

$n = 2: \ \{01, 10, 11\}$

$n = 3: \ \{010, 011, 101, 110, 111\} = \{011, 101, 111\} \sqcup \{010, 110\}$

Solution: For $n \geqslant 3$, break into cases, whether an admissible string ends in a 1 or a 0.

Bit strings

Find a recurrence relation and give initial conditions for the number of bit strings of length n that do not have two consecutive 0s. How many such bit strings are there of length five?

Examples:

$n = 1:\ \{0, 1\}$

$n = 2:\ \{01, 10, 11\}$

$n = 3:\ \{010, 011, 101, 110, 111\} = \{011, 101, 111\} \sqcup \{010, 110\}$

Solution: For $n \geqslant 3$, break into cases, whether an admissible string ends in a 1 or a 0.

Strings that end in a 1:

Bit strings

Find a recurrence relation and give initial conditions for the number of bit strings of length n that do not have two consecutive 0s. How many such bit strings are there of length five?

Examples:

$n = 1: \ \{0, 1\}$

$n = 2: \ \{01, 10, 11\}$

$n = 3: \ \{010, 011, 101, 110, 111\} = \{011, 101, 111\} \sqcup \{010, 110\}$

Solution: For $n \geqslant 3$, break into cases, whether an admissible string ends in a 1 or a 0.

Strings that end in a 1: you can take any admissible $n-1$ string, add a 1 to the end, and get an admissible n string.

Bit strings

Find a recurrence relation and give initial conditions for the number of bit strings of length n that do not have two consecutive 0s. How many such bit strings are there of length five?

Examples:

$n = 1: \ \{0, 1\}$

$n = 2: \ \{01, 10, 11\}$

$n = 3: \ \{010, 011, 101, 110, 111\} = \{011, 101, 111\} \sqcup \{010, 110\}$

Solution: For $n \geqslant 3$, break into cases, whether an admissible string ends in a 1 or a 0.

Strings that end in a 1: you can take any admissible $n - 1$ string, add a 1 to the end, and get an admissible n string.

$$\boxed{a_{n-1} \text{ of these}}$$

Bit strings

Find a recurrence relation and give initial conditions for the number of bit strings of length n that do not have two consecutive 0s. How many such bit strings are there of length five?

Examples:

$n = 1: \ \{0, 1\}$

$n = 2: \ \{01, 10, 11\}$

$n = 3: \ \{010, 011, 101, 110, 111\} = \{011, 101, 111\} \sqcup \{010, 110\}$

Solution: For $n \geqslant 3$, break into cases, whether an admissible string ends in a 1 or a 0.

Strings that end in a 1: you can take any admissible $n-1$ string, add a 1 to the end, and get an admissible n string.

$\boxed{a_{n-1} \text{ of these}}$

Strings that end in a 0: if an admissible string ends in a 0, then the second-to-last bit *has* to be a 1. So this falls into the first case, but for admissible strings of length $n-1$.

Bit strings

Find a recurrence relation and give initial conditions for the number of bit strings of length n that do not have two consecutive 0s. How many such bit strings are there of length five?

Examples:

$n = 1: \ \{0, 1\}$

$n = 2: \ \{01, 10, 11\}$

$n = 3: \ \{010, 011, 101, 110, 111\} = \{011, 101, 111\} \sqcup \{010, 110\}$

Solution: For $n \geqslant 3$, break into cases, whether an admissible string ends in a 1 or a 0.

Strings that end in a 1: you can take any admissible $n-1$ string, add a 1 to the end, and get an admissible n string.

$$\boxed{a_{n-1} \text{ of these}}$$

Strings that end in a 0: if an admissible string ends in a 0, then the second-to-last bit *has* to be a 1. So this falls into the first case, but for admissible strings of length $n-1$.

$$\boxed{a_{n-1-1} = a_{n-2} \text{ of these}}$$

Bit strings

Find a recurrence relation and give initial conditions for the number of bit strings of length n that do not have two consecutive 0s. How many such bit strings are there of length five?

Examples:

$n = 1:\ \{0, 1\}$

$n = 2:\ \{01, 10, 11\}$

$n = 3:\ \{010, 011, 101, 110, 111\} = \{011, 101, 111\} \sqcup \{010, 110\}$

Solution: For $n \geqslant 3$, break into cases, whether an admissible string ends in a 1 or a 0.

Strings that end in a 1: you can take any admissible $n-1$ string, add a 1 to the end, and get an admissible n string.

$$\boxed{a_{n-1} \text{ of these}}$$

Strings that end in a 0: if an admissible string ends in a 0, then the second-to-last bit *has* to be a 1. So this falls into the first case, but for admissible strings of length $n-1$.

$$\boxed{a_{n-1-1} = a_{n-2} \text{ of these}}$$

Rec. rel.: $a_n = a_{n-1} + a_{n-2}$

Bit strings

Find a recurrence relation and give initial conditions for the number of bit strings of length n that do not have two consecutive 0s. How many such bit strings are there of length five?

Examples:

$n = 1 : \ \{0, 1\}$

$n = 2 : \ \{01, 10, 11\}$

$n = 3 : \ \{010, 011, 101, 110, 111\} = \{011, 101, 111\} \sqcup \{010, 110\}$

Solution: For $n \geqslant 3$, break into cases, whether an admissible string ends in a 1 or a 0.

Strings that end in a 1: you can take any admissible $n-1$ string, add a 1 to the end, and get an admissible n string.

$$\boxed{a_{n-1} \text{ of these}}$$

Strings that end in a 0: if an admissible string ends in a 0, then the second-to-last bit *has* to be a 1. So this falls into the first case, but for admissible strings of length $n-1$.

$$\boxed{a_{n-1-1} = a_{n-2} \text{ of these}}$$

Rec. rel.: $a_n = a_{n-1} + a_{n-2}$, **Init. conds.:** $a_1 = 2$, $a_2 = 3$.

Codeword enumeration

A computer system considers a string of decimal digits a "valid codeword" if it contains an even number of 0 digits. For example,

1230407869 is valid, but 120987045608 is not.

Codeword enumeration

A computer system considers a string of decimal digits a "valid codeword" if it contains an even number of 0 digits. For example,

1230407869 is valid, but 120987045608 is not.

Let a_n be the number of valid n-digit codewords. Find a recurrence relation and initial conditions for a_n.

Codeword enumeration

A computer system considers a string of decimal digits a "valid codeword" if it contains an even number of 0 digits. For example,

1230407869 is valid, but 120987045608 is not.

Let a_n be the number of valid n-digit codewords. Find a recurrence relation and initial conditions for a_n.

Solution: For $n \geqslant 2$, break into cases, whether an admissible string ends in a 0 or not.

Codeword enumeration

A computer system considers a string of decimal digits a "valid codeword" if it contains an even number of 0 digits. For example,

1230407869 is valid,   but 120987045608 is not.

Let a_n be the number of valid n-digit codewords. Find a recurrence relation and initial conditions for a_n.

Solution: For $n \geqslant 2$, break into cases, whether an admissible string ends in a 0 or not.

Strings that do not end in a 0:

Codeword enumeration

A computer system considers a string of decimal digits a "valid codeword" if it contains an even number of 0 digits. For example,

1230407869 is valid,   but 120987045608 is not.

Let a_n be the number of valid n-digit codewords. Find a recurrence relation and initial conditions for a_n.

Solution: For $n \geqslant 2$, break into cases, whether an admissible string ends in a 0 or not.

Strings that do not end in a 0: the first $n-1$ numbers are also valid.

Codeword enumeration

A computer system considers a string of decimal digits a "valid codeword" if it contains an even number of 0 digits. For example,

1230407869 is valid, but 120987045608 is not.

Let a_n be the number of valid n-digit codewords. Find a recurrence relation and initial conditions for a_n.

Solution: For $n \geqslant 2$, break into cases, whether an admissible string ends in a 0 or not.

Strings that do not end in a 0: the first $n-1$ numbers are also valid.

$$\boxed{9 * a_{n-1} \text{ of these}}$$

Codeword enumeration

A computer system considers a string of decimal digits a "valid codeword" if it contains an even number of 0 digits. For example,

1230407869 is valid, but 120987045608 is not.

Let a_n be the number of valid n-digit codewords. Find a recurrence relation and initial conditions for a_n.

Solution: For $n \geqslant 2$, break into cases, whether an admissible string ends in a 0 or not.

Strings that do not end in a 0: the first $n-1$ numbers are also valid.

$$\boxed{9 * a_{n-1} \text{ of these}}$$

Strings that end in a 0:

Codeword enumeration

A computer system considers a string of decimal digits a "valid codeword" if it contains an even number of 0 digits. For example,

1230407869 is valid, but 120987045608 is not.

Let a_n be the number of valid n-digit codewords. Find a recurrence relation and initial conditions for a_n.

Solution: For $n \geqslant 2$, break into cases, whether an admissible string ends in a 0 or not.

Strings that do not end in a 0: the first $n-1$ numbers are also valid.

$$\boxed{9 * a_{n-1} \text{ of these}}$$

Strings that end in a 0: the first $n-1$ numbers are *not* valid.

Codeword enumeration

A computer system considers a string of decimal digits a "valid codeword" if it contains an even number of 0 digits. For example,

1230407869 is valid, but 120987045608 is not.

Let a_n be the number of valid n-digit codewords. Find a recurrence relation and initial conditions for a_n.

Solution: For $n \geqslant 2$, break into cases, whether an admissible string ends in a 0 or not.

Strings that do not end in a 0: the first $n-1$ numbers are also valid.

$$\boxed{9 * a_{n-1} \text{ of these}}$$

Strings that end in a 0: the first $n-1$ numbers are *not* valid.

$$\boxed{10^{n-1} - a_{n-1} \text{ of these}}$$

Codeword enumeration

A computer system considers a string of decimal digits a "valid codeword" if it contains an even number of 0 digits. For example,

1230407869 is valid, but 120987045608 is not.

Let a_n be the number of valid n-digit codewords. Find a recurrence relation and initial conditions for a_n.

Solution: For $n \geqslant 2$, break into cases, whether an admissible string ends in a 0 or not.

Strings that do not end in a 0: the first $n-1$ numbers are also valid.

$$\boxed{9 * a_{n-1} \text{ of these}}$$

Strings that end in a 0: the first $n-1$ numbers are *not* valid.

$$\boxed{10^{n-1} - a_{n-1} \text{ of these}}$$

Rec. rel.: $a_n = 9 * a_{n-1} + (10^{n-1} - a_{n-1})$

Codeword enumeration

A computer system considers a string of decimal digits a "valid codeword" if it contains an even number of 0 digits. For example,

1230407869 is valid, but 120987045608 is not.

Let a_n be the number of valid n-digit codewords. Find a recurrence relation and initial conditions for a_n.

Solution: For $n \geqslant 2$, break into cases, whether an admissible string ends in a 0 or not.

Strings that do not end in a 0: the first $n-1$ numbers are also valid.

$$\boxed{9 * a_{n-1} \text{ of these}}$$

Strings that end in a 0: the first $n-1$ numbers are *not* valid.

$$\boxed{10^{n-1} - a_{n-1} \text{ of these}}$$

Rec. rel.: $a_n = 9 * a_{n-1} + (10^{n-1} - a_{n-1}) = 8a_{n-1} + 10^{n-1}$

Codeword enumeration

A computer system considers a string of decimal digits a "valid codeword" if it contains an even number of 0 digits. For example,

1230407869 is valid, but 120987045608 is not.

Let a_n be the number of valid n-digit codewords. Find a recurrence relation and initial conditions for a_n.

Solution: For $n \geqslant 2$, break into cases, whether an admissible string ends in a 0 or not.

Strings that do not end in a 0: the first $n-1$ numbers are also valid.

$$\boxed{9 * a_{n-1} \text{ of these}}$$

Strings that end in a 0: the first $n-1$ numbers are *not* valid.

$$\boxed{10^{n-1} - a_{n-1} \text{ of these}}$$

Rec. rel.: $a_n = 9 * a_{n-1} + (10^{n-1} - a_{n-1}) = 8a_{n-1} + 10^{n-1}$,

Initial conds.: $a_1 = 9$.

Codeword enumeration

A computer system considers a string of decimal digits a "valid codeword" if it contains an even number of 0 digits. For example,

1230407869 is valid, but 120987045608 is not.

Let a_n be the number of valid n-digit codewords. Find a recurrence relation and initial conditions for a_n.

Solution: For $n \geqslant 2$, break into cases, whether an admissible string ends in a 0 or not.

Strings that do not end in a 0: the first $n-1$ numbers are also valid.

$$\boxed{9 * a_{n-1} \text{ of these}}$$

Strings that end in a 0: the first $n-1$ numbers are *not* valid.

$$\boxed{10^{n-1} - a_{n-1} \text{ of these}}$$

Rec. rel.: $a_n = 9 * a_{n-1} + (10^{n-1} - a_{n-1}) = 8a_{n-1} + 10^{n-1}$,

Initial conds.: $a_1 = 9$.

You try: Exercise 30