Algorithms

Karina Kudymova, 2013

How to make a cup of tea

Get a kettle

Fill the kettle with water

Boil the water

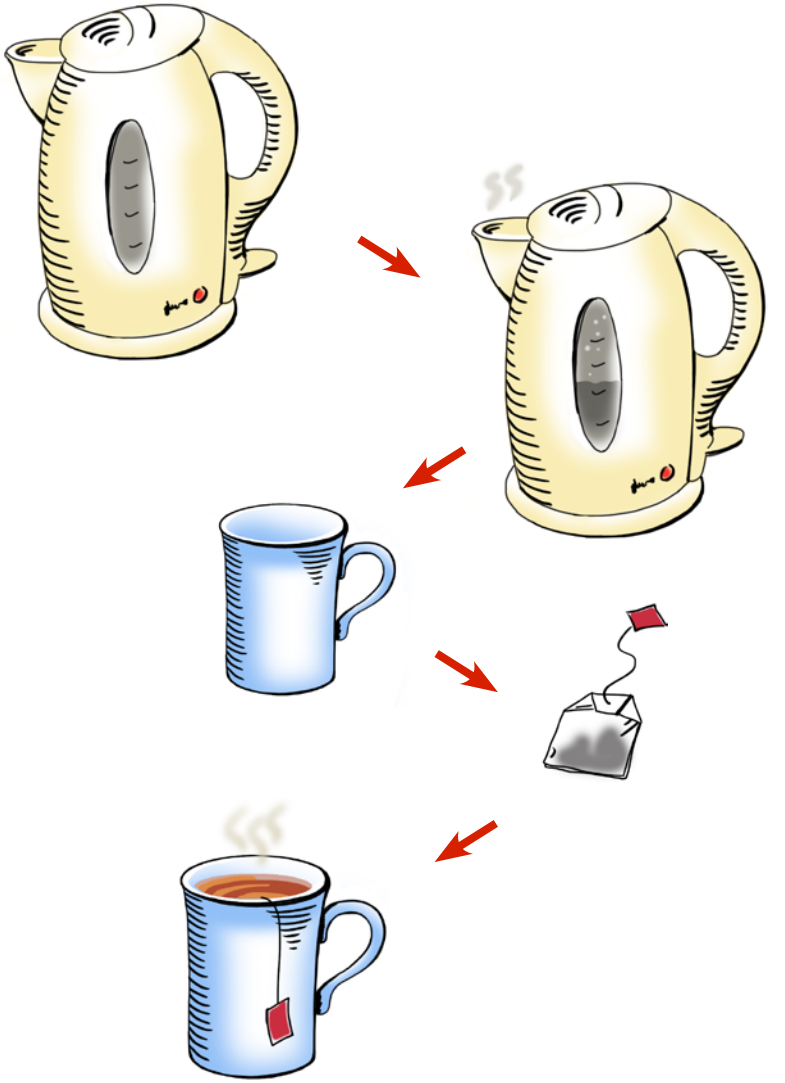
Get a cup

Pour hot water into the cup

Get a tea bag

Put the tea bag into hot water

Wait a few minutes

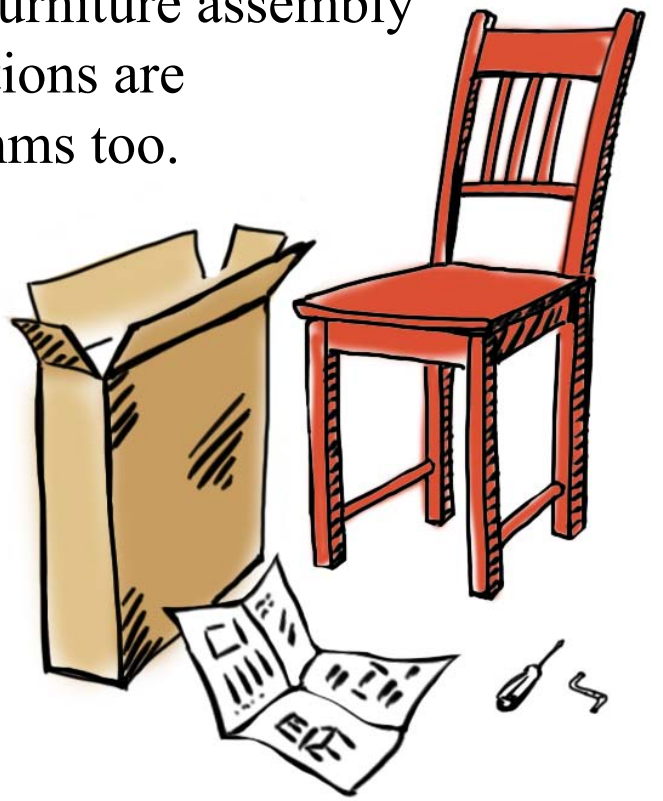


What we just created is an algorithm of making a cup of tea

In general, an *algorithm* is a list of instructions for accomplishing some task.

Recipes in a cookbook are examples of algorithms.

IKEA furniture assembly instructions are algorithms too.

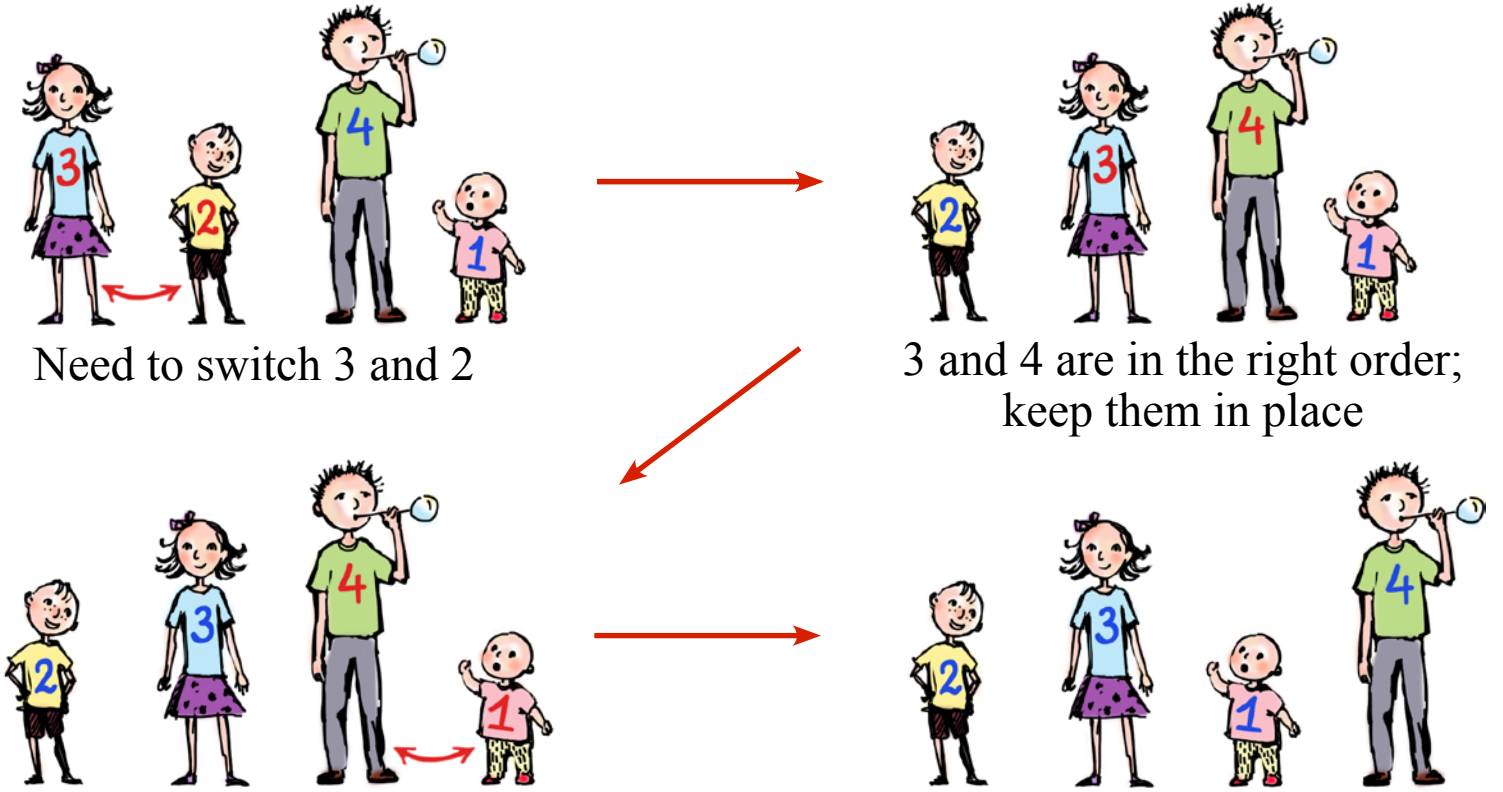


All computer programs are formalized algorithms

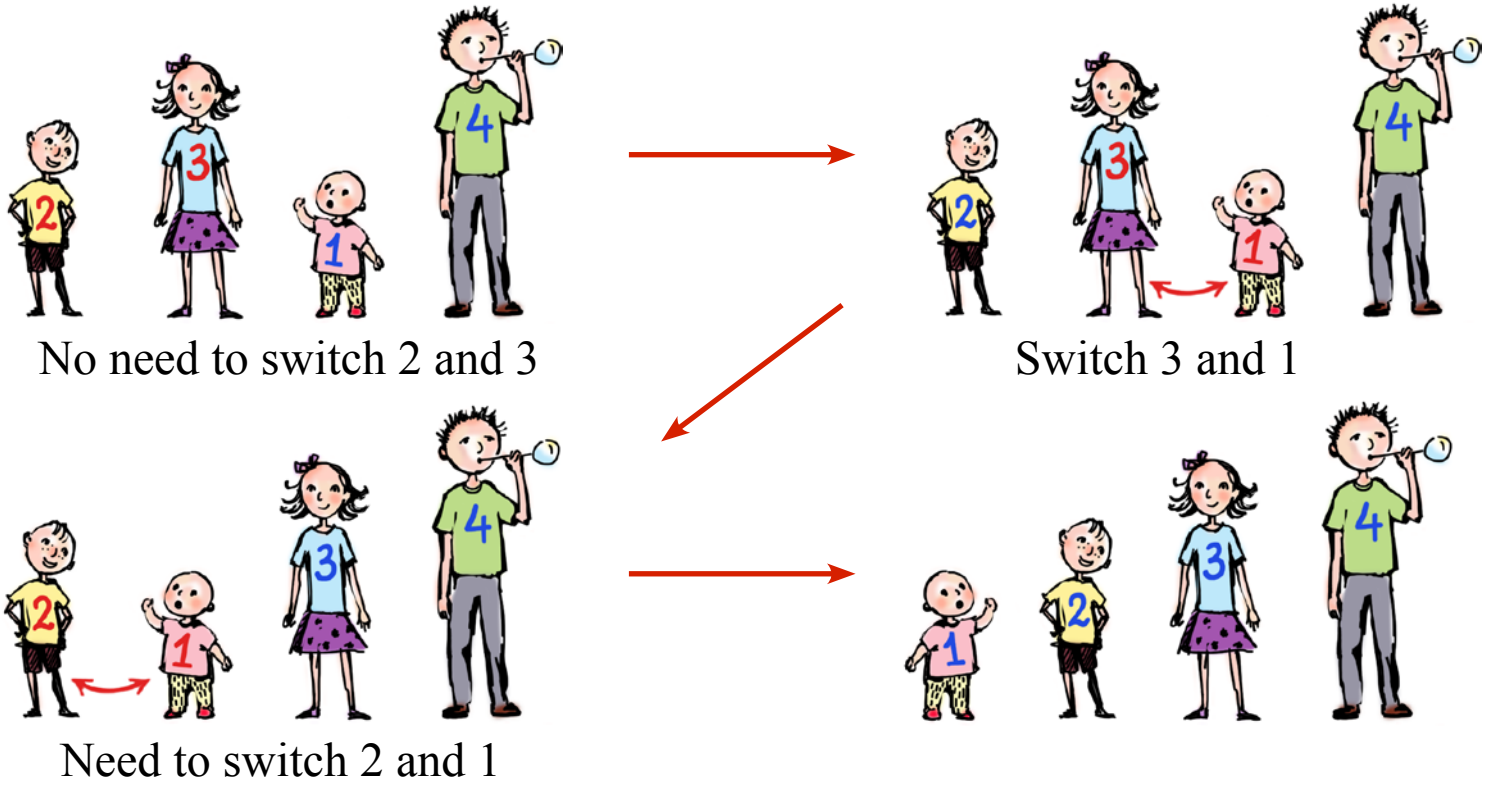
```
for(i=0; i<N; i++){
  for(j=0; j<i; j++){
    if(a[j]>a[j+1])
swap(&a[j],&a[j+1]);
  }
}
```

Let's look at a simple sorting algorithm. We will arrange kids by height from the shortest to the tallest.

First we'll compare the first and the second kid and if they are standing in the wrong order we'll ask them to switch places. Then we'll compare the the second and the third and switch them if necessary. We'll keep doing it all the way up to the last pair. We won't be done sorting then, but the tallest kid will be standing on the right. (Can you tell why?)



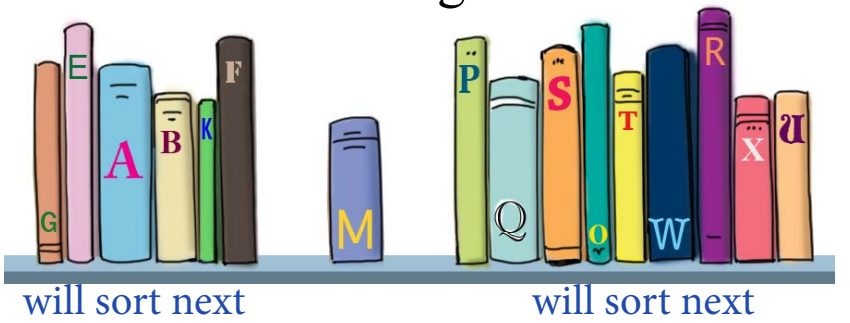
In our example the tallest kid moved to the right, now we need to sort the remaining ones. We'll apply the same procedure once again so the second tallest child moves to the right place. We don't have to compare the last pair because the tallest person is already on the right. We can repeat the procedure each time reducing the number of comparisons by one.



This algorithm is called bubble sort, because it resembles the way water bubbles rise in water. The bigger the bubble, the faster it rises.

Bubble sort is not very efficient: it takes a lot of swapping to get things sorted. Another algorithm is quick sort. It is named that way because usually it requires much fewer operations to complete the sort.

Suppose you need to arrange books on a shelf in alphabetical order. Pick a random book and put all the books that should come before it to its left and the ones after it to its right.



The books are not sorted yet, but we now have two shorter sequences of books to sort. We can apply the same procedure to the two halves (pick a book and move remaining books to its left and right). Then we'll apply it to quarters, eighth, etc. until we are done.

We just saw an example of a recursive algorithm. Recursive algorithms break down a problem into smaller or simple pieces and apply the same algorithm to them.

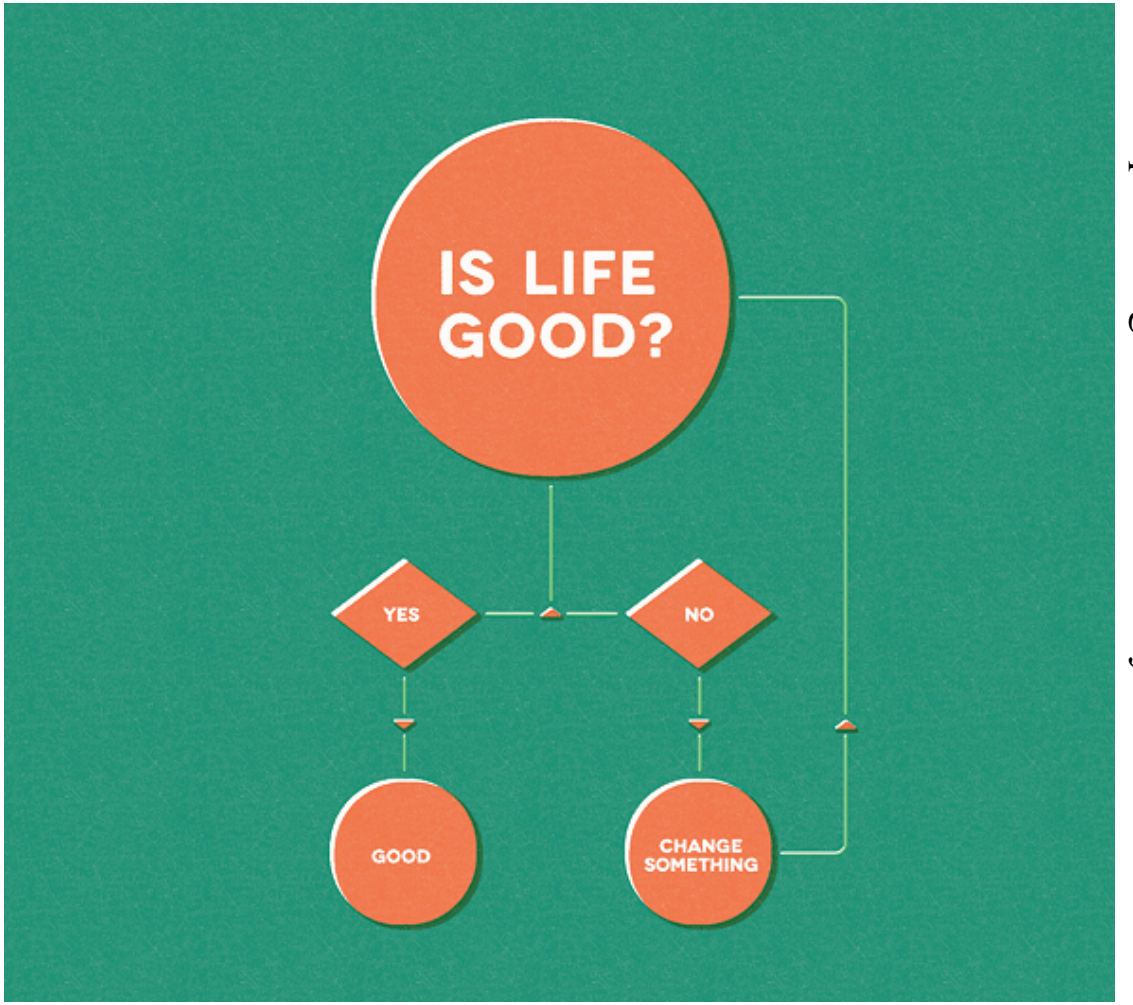
Here is a recursive algorithm to compute n! = 1x2x3 ... x(n-1)xn

To compute n!:
- If n is 1, then n!=1
- Otherwise, compute (n-1)! and multiply by n.

```
int factorial(int n){
  if(n==1) return(1);
  else return(n*factorial(n-1));
}
```

Recursive, adj.: see Recursive
(from The Computer Contradictionary
by Stan Kelly-Bootle)

Algorithms can be often represented with flow charts or diagrams. Here is a simple example.



From http://islifegood.tumblr.com by Gustavo Vieira