Backtracking

Backtracking is a method of solving problems by making a **series of choices** that we can return or ***backtrack*** to.

- A backtracking algorithm tries to construct a problem ***incrementally***, one piece at a time.
  - When the algorithm needs to decide between multiple alternatives to the next component, it recursively evaluates every alternative and then chooses the best one.

Analogy:

It's like when you lose your keys at home and you have to try and go to find them. You'll first panic but soon come to the conclusion that they're definitely somewhere in your home. So you go all across your place checking every possible place you could've previously left them. At each decision point you have other possible points for you to explore until you find your lost keys.

In this analogy you have the 3 fundamental keys that identifies a backtracking problem.

1. First, you decided your **decision space** by suspecting that your keys are somewhere in your home. Your decision space is your range of choices.
2. Second, you have constraints, your problem comes with set constraints. You must check each sensible place you could've lost them room by room. This means you won't be sticking your head in the oven or dunking your hands into the toilet bowl. There are possible points of your home that we've ruled out.
3. You have a goal, your goal is finding your lost keys.

The 3 Fundamentals of Backtracking

Theese are the basics that look like the following:

1. You make **choices**.
2. You have **constraints** on those choices.
3. And you have a **goal** to make on those choices.

When backtracing you explore a decision space, perform on each subproblem, adhere to the constraints of the problem, and converge to a base case.

But not every decision will work out. In which case, we will backtrack and eject our failed decision. In other words, we simply **undo** it. We xplore and recurse so long as the decision works.

Key signs that a question requires backtracking:

If a question says *generate all* or *compute all* then it is an ***exhaustive*** problem which often times are problems that can be solved using backtracking.