Noah Parker
Chapter 13 Study Sheet

Triangle numbers How to check if a number is Triangular? The idea is based on the fact that n'th triangular number can be written as sum of n natural numbers, that is n*(n+1)/2. The reason for this is simple, base line of triangular grid has n dots, line above base has (n-1) dots and so on.
This example it find the triangular number with any natural range of numbers.

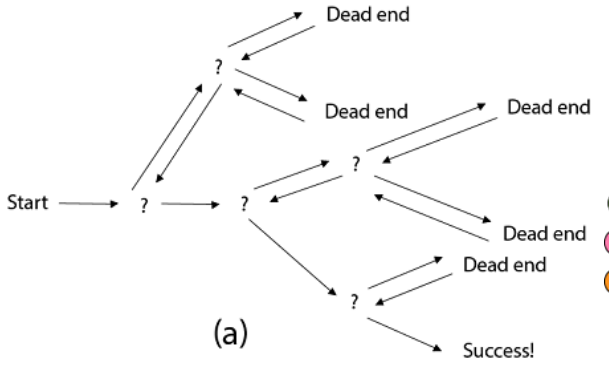
```
public class TriangularNumbers {
  public static void main(String[] args) {
    int starting_number = 1;
    int ending_number = 10;
    System.out.println("List of Triangular Numbers ");
    for (int i = starting_number; i <= ending_number; i++) {
      int triangular = 0;
      for (int j = 1; j <= i; j++) {
        triangular = triangular + j;
      }
      System.out.println(i + " = " + triangular);}}}
```

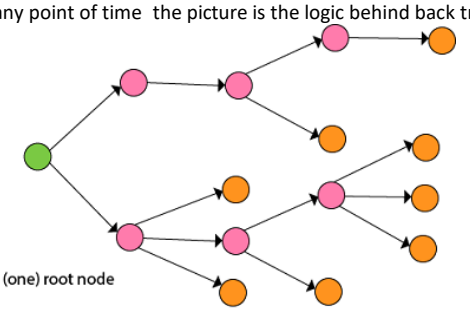
Recursive Helper Methods Sometimes it is easier to find a recursive solution if you change the original problem slightly. Then the original problem can be solved by calling a recursive helper method. The Efficiency of Recursion. Recursion in java is a process in which a method calls itself continuously. A method in java that calls itself is called recursive method. It makes the code compact but complex to understand.
The example top the right is showing how to use recursion to solve factorial numbers.
Below is how the program To the right Works

```
public class RecursionExample {
  static int factorial(int n){
      if (n == 1)
        return 1;
      else
        return(n * factorial(n-1));
  }
public static void main(String[] args) {
System.out.println("Factorial of 5 is: "+factorial(5));
} }
Output = Factorial of 5 is: 120
```

```
factorial(5)
  factorial(4)
    factorial(3)
      factorial(2)
        factorial(1)
          return 1
        return 2*1 = 2
      return 3*2 = 6
    return 4*6 = 24
  return 5*24 = 120
```

Permutations A permutation of a set is a rearrangement of its elements. A set which consists of n elements has n! permutations. Here n! is the factorial, which is the product of all positive integers smaller or equal to n.
Mutual recursion Two functions are said to be mutually recursive if the first calls the second, and in turn the second calls the first.
Backtracking is an algorithmic-technique for solving problems recursively by trying to build a solution incrementally, one piece at a time, removing those solutions that fail to satisfy the constraints of the problem at any point of time the picture is the logic behind back tracking




Backtracking algorithm determines the solution by systematically searching the solution space for the given problem. Backtracking is a depth-first search with any bounding function. All solution using backtracking is needed to satisfy a complex set of constraints. The constraints may be explicit or implicit.Explicit Constraint is ruled, which restrict each vector element to be chosen from the given set.
Implicit Constraint is ruled, which determine which each of the tuples in the solution space, satisfy the criterion function.

To "explore" node N:
1. If N is a goal node, return "success"
2. If N is a leaf node, return "failure"
3. For each child C of N,
   Explore C
   If C was successful, return "success"
4. Return "failure"
This goes with the C picture both a and c picture came from:
https://www.javatpoint.com/backtracking-introduction