<u>Visualization</u>

1. Introduction

This visualization can visualize the recursion tree of a recursive algorithm.
But you can also visualize the Directed Acyclic Graph (DAG) of a DP algorithm.

2. Recursion Tree/DAG

This is the Recursion Tree/DAG visualization area.
Note that due to combinatorial explosion, it will be very hard to visualize Recursion Tree for large instances.
And for Recursion DAG, it will also very hard to minimize the number of edge crossings in the event of overlapping subproblems.

3. Examples

Select one of the examples, or write your own code.
Note that the visualization can run *any* javascript code, including malicious code, so please be careful.
Click the 'Run' button to start the visualization after you have selected or written a valid JavaScript code!

4. Factorial Example

The Factorial example computes the factorial of a number **N**.
It is one of the simplest (tail) recursive function that can actually be rewritten into iterative version.

5. Fibonacci example

The Fibonacci example computes the **N**-th Fibonacci number.
Unlike Factorial example, this time each recursive step recurses to two other smaller sub-problems. It can still be written in iterative fashion after one understands the concept of Dynamic Programming. Fibonacci recursion tree (and DAG) are frequently used to showcase the basic idea of recursion.

6. Catalan example

The Catalan example computes the **N**-th catalan number recursively.

7. GCD example

The GCD example computes the Greatest Common Divisor of two numbers **A** and **B** recursively.

8. N Choose K

The N Choose K computes the binomial coefficient C(**N**, **K**).

9. Range Sum Query example

The Range Sum Query example computes the maximum value of S(l,r), where S(l,r) = a1[l] + a1[l+1] + ... + a1[r], where 1≤l≤r≤i.

10. Knapsack example

The Knapsack example solves the 0/1 Knapsack Problem: What is the maximum value that we can get, given a knapsack that can hold a maximum weight of w, where the value of the i-th item is a1[i], the weight of the i-th item is a2[i]?

11. Coin Change example

The Coin Change example solves the Coin Change problem: Given a list of coin values in a1, what is the minimum number of coins needed to get the value v?

12. Longest Increasing Subsequence example

The Longest Increasing Subsequence example solves the Longest Increasing Subsequence problem: Given an array a1, how long is the Longest Increasing Subsequnce of the array?

13. Traveling Salesman example

The Traveling Salesman example solves the Traveling Salesman Problem on small graph: How long is the shortest path that goes from city 0, passes through every city once, and goes back again to 0? The distance between city i and city j is denoted by a1[i][j].

14. Matching problem

The Matching problem computes the maximum number of matching on a small graph, which is given in the adjacency matrix a1.