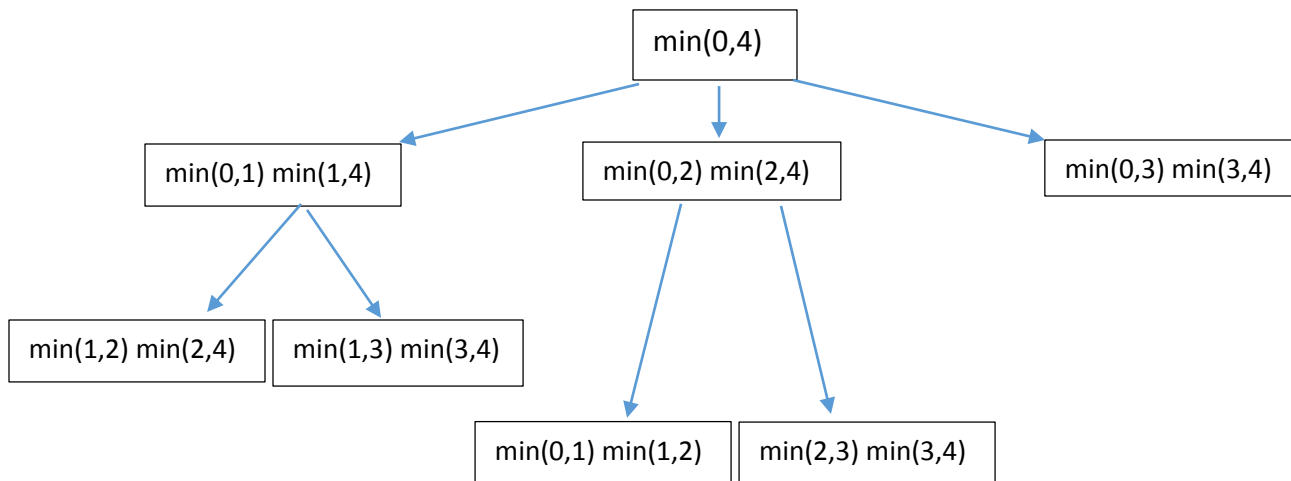


## CO322: Data Structure and Algorithms

### Lab2: Dynamic Programming

- 2) minCost function runs every possible path from 0 to N-1 times.  
So run-time complexity =  $O(N!)$ .



Like this pattern will continue.

- 3) In the recursion method sometimes same value find for multiple times. It takes a lot of time. If we use some memory to keep these values and then we can get rid of finding the same value again and again. Then we can use the stored values from the memory.
- 5) According to this example the idea is to first calculate min cost for station 1, then for station 2, and so on. These costs are stored in an array `dis[0...N-1]`.
- 1) The min cost for station 0 is 0  $\rightarrow$  `dis[0] = 0`
  - 2) The min cost for station 1 is  $\rightarrow$  `cost[0][1]`, `dis[1] = cost[0][1]`
  - 3) The min cost for station 2 is minimum of following two.
    - a) `dis[0] + cost[0][2]`
    - b) `dis[1] + cost[1][2]`
  - 3) The min cost for station 3 is minimum of following three.
    - a) `dis[0] + cost[0][3]`

b)  $\text{dis}[1] + \text{cost}[1][3]$

c)  $\text{dis}[2] + \text{cost}[2][3]$

And we can find for station 4 as well.

Then number of comparisons is  $N(N-2)/2$ .

So complexity of run time is  $O(N^2)$ .