CO 322 Data Structures and Algorithms

Lab 02 - Dynamic Programming

Name : A.L.V.H.Dharmathilaka

Registration number : E/16/086

(b) What is the runtime complexity of your implementation .

In the recursive implementation same sub problem is always recomputed as it always checks possible path from 0 to given station. So the time complexity of the above implementation is exponential.

Using the function minCost(int FromStation, int ToStation)

From station	To station	n	Time complexity
0	1	1	1
0	2	2	3
0	3	3	9
0	4	4	27
0	5	5	81
0	6	6	243

So the time complexity is $T(n)=3^{(n-1)}$

From big O notation it will be **$O(3^n)$**

(c) Argue that dynamic programming can be used to improve the runtime.

Whenever the repeated calls for same outputs is done in recursive solution, it can be optimized to a dynamic programming implementation. The idea is to store the results of the sub problems and use them when again another sub problems is calling. So no need of re-computing them. In dynamic programming each sub problem is solved just once. As a result of it, the time complexity of the implementation is decreased.

In this task1 implementation every sub problem tries every possible path from 0 to given station. That means it checks earlier checked sub problem so that it increases runtime. If we store previously calculated cost then there is no time allocation for solving those problems in next time but to store sub problem values, it consumes sometime but most of the times it will be less than time taken to solve that sub problem again.

(e) Calculate the runtime of your implementation in part 4 above. Assume, hashing is O(1).

Using the function minCostDynamicPro(int FromStation, int ToStation)

From station	To station	(n)	Time complexity
0	1	1	1
0	2	2	3
0	3	3	7
0	4	4	13
0	5	5	21
0	6	6	31

Complexity = $n^2 - n + 1$

So the runtime will be **$O(n^2)$**