



AFIFA TAZREMIN OISHI

ID : 19-40028-1
SECTION : E

Date of submission :8th September ,2020

Algorithm Final Assignment Part – 2

Set 01

Answer to the question no:1

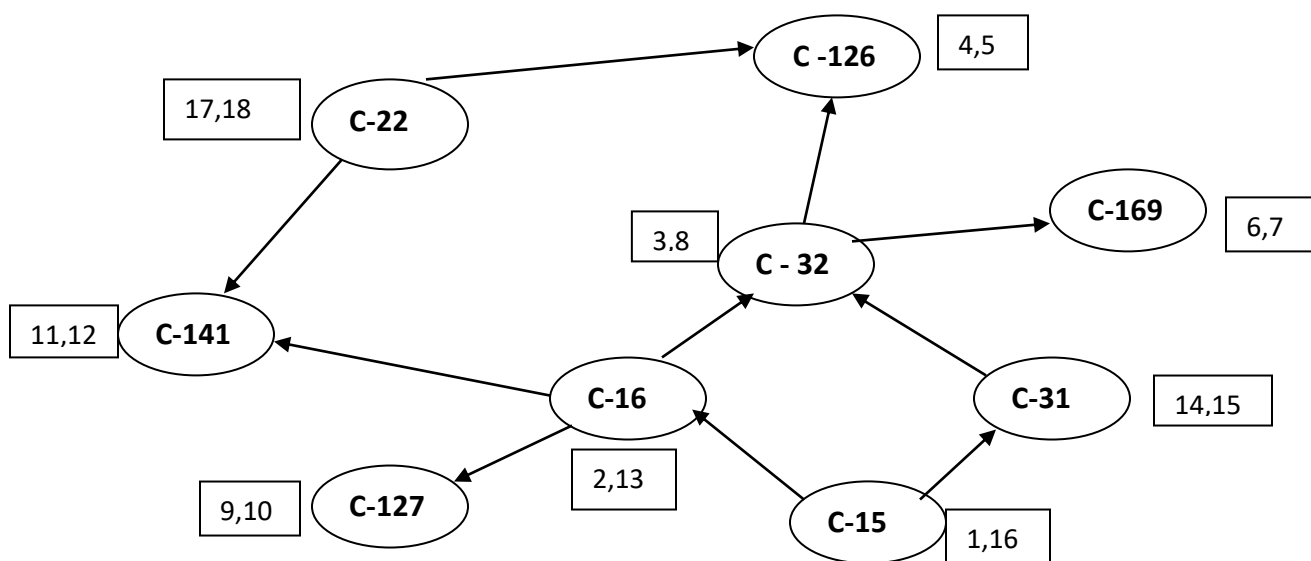
Herry will face some problem to take new courses because there are some prerequisite courses behind the particular courses. So Herry should take a process that's why he can complete his prerequisite.

Herry can solve this problem by graph algorithm. He can choose Depth-First Search(DFS) algorithm to solve the problem. It is a directional graph so in this process he can give a clear view to take his courses.

The idea behind DFS is to go as deep into the graph as possible, and backtrack once we are at a vertex without any unvisited adjacent vertices. It is very easy to describe / implement the algorithm recursively: We start the search at one vertex. After visiting a vertex, we further perform a DFS for each adjacent vertex that we haven't visited before. This way we visit all vertices that are reachable from the starting vertex.

Depth First Search will also find the shortest paths in a tree (because there only exists one simple path), but on general graphs this is not the case. So Herry can use DFS algorithm to get a perfect solution.

Answer to the question no: 2



Topological sorting:

Arrange the nodes according to the finish time:

Course 22(18)

Course 15(16)

Course 31(15)

Course 16(13)

Course 141(12)

Course 127(10)

Course 32(8)

Course 169(7)

Course 126(5)

In this graph we can see many courses has no prerequisite. So Herry should take the courses first.

Answer to the question no:3

The reason I use Topological sort algorithm is Topological sort gives a linear ordering of vertices in a *directed acyclic graph* such that, for every directed edge $a \rightarrow b$, vertex 'a' comes before vertex 'b'. As we know that the source vertex will come after the destination vertex, so we need to use a stack to store previous elements. After completing all nodes, we can simply display them from the stack.

The time complexity of this program is $O(V+E)$. The above algorithm is simply DFS with an extra stack. So time complexity is the same as DFS.

Answer to the question no:4

Topological sort is the best solution for the above problem. We can not use greedy approach because greedy approach are basically use for fractional number. Greedy is an algorithmic paradigm that builds up a solution piece by piece, always choosing the next piece that offers the most obvious and immediate benefit. But the above problem is not like that so we can not use greedy approach.

Dynamic programming (DP) is an algorithmic technique for solving an optimization problem by breaking it down into simpler subproblems and utilizing the fact that the optimal solution to the overall problem depends upon the optimal solution to its subproblems. But the above problem is a graph mapping problem. We basically use dynamic programming for MCM, LCM, 0-1 Knapsack problem. So we can not use dynamic programming also to solve the above problem.

-----O-----