DYNAMIC PROGRAMMING

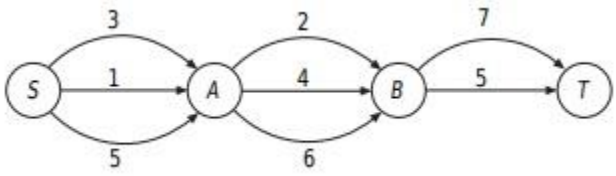
The dynamic programming strategy is a very useful technique to solve many combinatorial optimization problems. Before presenting the dynamic programming method, it may be convenient to reconsider a typical case where the greedy method can be applied.

Example:

Suppose you want to find the shortest route from S to T. In this case, the shortest route can be found by the following line of reasoning:

1.Because it is known that the shortest route must pass through vertex A, it is necessary to find a shorter route from S to A. The cost of this route is 1.
2. Because it is known that the shortest route must pass through B, there is a shorter route from A to B. The cost of this route is 2.
3. Similarly, there is a shorter route from B to T whose cost is 5.

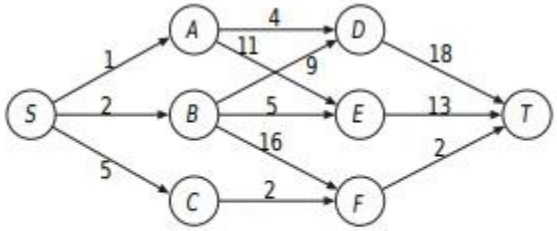
A case in which the greedy method works:



(figure a)

Thus, the total cost of the shortest route from S to T is 1 + 2 + 5 = 8. Also, this the shortest route problem was actually solved by applying a greedy method. why is it possible to use the greedy method to solve that problem? This is possible because it is definitely known that the solution must consist of a sub-route from S to A, a sub-route from A to B, etc. Consequently, our strategy for solving the problem first finds a shorter sub-path from S to A, then a shorter sub-path from A to B, and so on.

Below is a case in which the greedy method does not work. Consider figure b. Again, we want to find a shorter route from S to T. However, this time we do not know which of the vertices A, B and C must pass the shortest route. If the greedy method is used to solve this problem, the vertex A is chosen because the cost associated with the edge from S to A is the lowest. After A is chosen, D is chosen. The cost of this route is 1 + 4 + 18 = 23. This is not the shortest route because the shortest route is S -> C -> F -> T, whose total cost is 5 + 2 + 2 = 9.



$S \xrightarrow{5} C \xrightarrow{2} F \xrightarrow{2} T$

(figure b)

CONCEPTUAL MAP