cpt_s 350

Homework 8

11641327 Yu-Chieh Wang

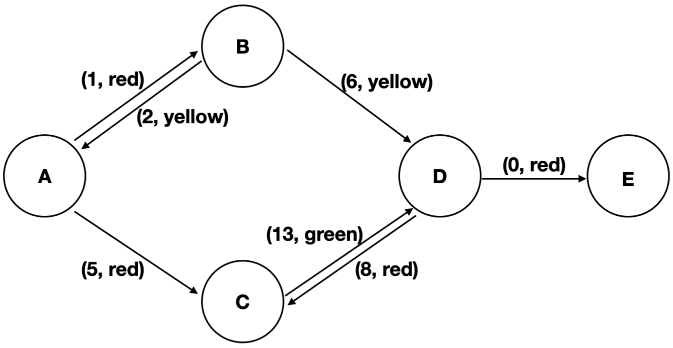
2020/4/16

Given a directed graph with weight and color on each edge.

1. For a given number k, enumeration the first i-th shortest paths, for all $1 \leq i \leq k$, from the initial to the final.

    To get the first shortest path, and the edges(i) it caress must $\leq k$, we need to run the graph as a tree structure, because every directed edge will lead us to a different leaf. For example, we make all edges as a table, then use code to run it.

u(From)	v(To)
A	B
A	C
B	A
B	D
C	D
D	C
D	E



If we want to get a path from node A to node E, we have to do dynamic process as following:

	A	B	C	D	E
A	0	1	1	1	1
B	1	1	1	2	2
C	1	1	1	2	2
D	1	1	2	2	3
E	1	1	2	2	3

Then, to check if the last number is less or equal to k.

If so, return it.

Else, we need to modify the sequence and then run it the dynamic program again.

2. Finding a shortest path that does not have a red edge immediately followed by a yellow edge.

   First of all, we need to get all possible paths without any loops.

   Second, remove the paths with any red edge.

   Finally, return the shortest path.

3. For each path w from the initial to the final, one can collect the colors on the path and therefore, a color sequence $c(w)$ is obtained. Notice that, it might be the case that two distinct paths w and w' corresponds to the same color sequence; I.e., $c(w) = c(w')$. Computing the size of the set $\{c(w) : w$ is a path from the initial to the final$\}$.

   There are two situations:

   First, all possible paths don't contain any loop, so the lengths of the paths are finite.

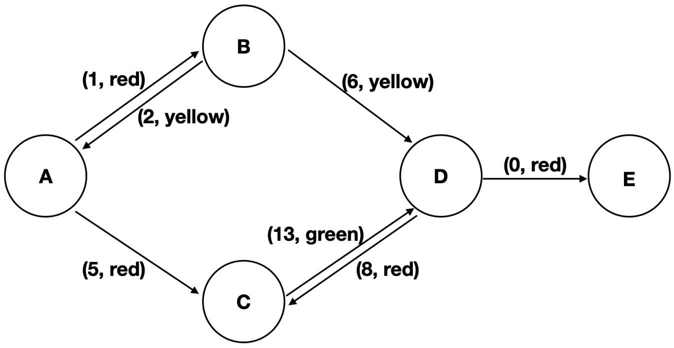
   Therefor, we can turn the paths to color paths and then compute the size of the set.

   On the other hand, if any paths contain loops, the graph may run forever, and we will get infinite possible paths, so it's impossible to compute the size of the set.

4. For each path w from the initial to the final, one can multiply the weights on the path and therefore, a number $W(w)$ is obtained. Find a path w from the initial to the final such that $W(w)$ is minimal.

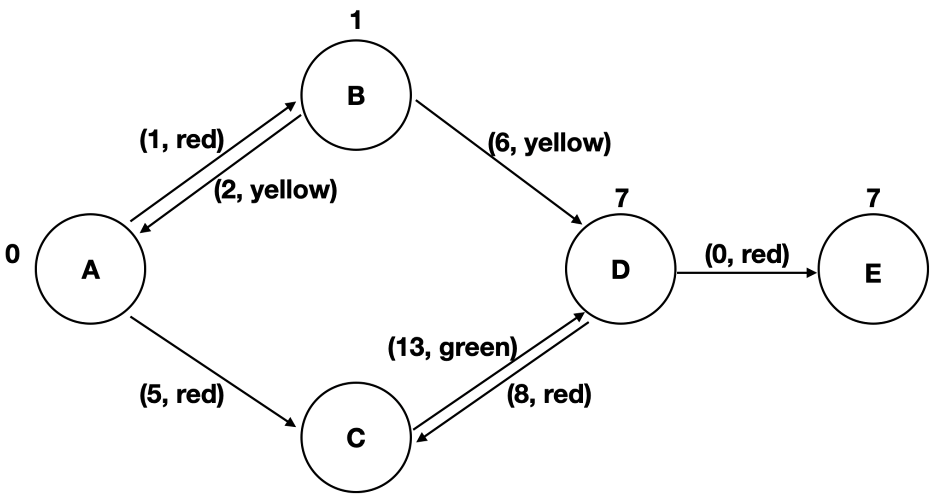
   To answer this question, we need to run the shortest path algorithm. To run the algorithm, we must count the minimal number of each node.

Using previous example:



At the beginning, the node A is 0. Then we chose a path B or C.

If we choose B, the number of node B will become 1. Then, we go to the next node D, the number of node D will become 7. Finally, we go to the node E, then we get the number from node A to node E is 7.



However, there is one node left, so we need to run another path. From A to C, we can get the number of node C is 5. If we keep go to the next node D, D suppose to get the number 18, but our purpose is to get the minimal number of each node, so we ignore it and won't go to the next step. In the end, when every node gets a number, the process of running the algorithm is done.