**Ch. 6.4-5 Graph**

1. Many shortest path algorithms are based on two important concepts, INITIALIZE-SINGLE-SOURCE and RELAX. Please explain how RELAX works following the way INITIALIZE-SINGLE SOURCE is explained in the following.

|  |
| --- |
| INITIALIZE-SINGLE-SOURCE(G,s)  for each vertex v belongs to G.v  dist[v] = infinity  predecessor[v] = NIL  dist[s] = 0 |
| RELAX(u,v,length) **//length(u,v) means edge length of (u,v)**  If dist[v] > dist[u]+length(u,v)  dist[v] = dist[u]+length(u.v)  predecessor[v]= u |

< Explain the Concept of INITIALIZE-SINGLE-SOURCE >

(1) Compute the shortest path to all vertices in the input graph G starting from the single source vertex s.

(2) In the for loop, initialize the distance from the source s to each vertex v in the graph G to be infinity so that later any path with shorter length can be recorded. At the same time set the predecessor of the vertex v under processing to be NIL, i.e. predecessor[v]= NIL, so that later this field can record the predecessor in the shortest path of the vertex v vertex.

(3) Initialize the distance to the source node s to be zero, since the shortest length to the source node itself is 0.

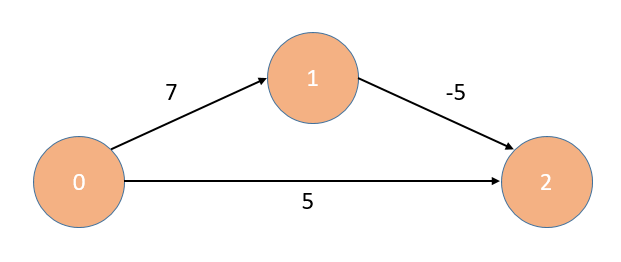
1. Read the following pseudocode for Dijkstra shortest path algorithm and then answer the questions.

|  |
| --- |
| void MatrixWDigraph::ShortestPath(const int n,const int v){      // dist[j], 0<=j<n, is set to the length of the shortest path from v to j      // in a digraph G with n vertices and edge lengths given by length[i][j]      for(int i=0;i<n;i++){          s[i]=false;          dist[i]=length[v][i];      }      s[v]=true;      dist[v]=0;      for(i=0;i<n-2;i++){  // determine n-1 paths from vertex v          int u=Choose(n); // choose returns a value u such that                           // dist[u]=minimum dist[w], where s[w]=false          s[u]=true;          for(int w=0;w<n;w++)              if(!s[w]&&dist[u]+length[u][w]<dist[w])                  dist[w]=dist[u]+length[u][w];      }  } |

(1) Explain how the pseudocode work.

(2) For the second for loop “for(i=0;i<n-2;i++)”, why index I i has to stop at n-2, not n?

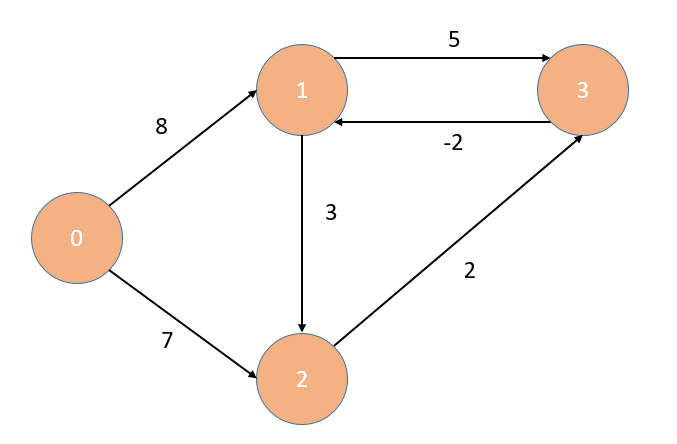
(3) When applying Dijkstra algorithm to the graph below starting from the vertex 0, what issue may occur?



1. In addition to the Dijkstra algorithm, BellmanFord algorithm can also solve the single-source shortest paths problem. A sample BellmanFord pseudocode is listed below for your reference.

TAs : Although the BellmanFord algorithm is in the self-study topic, it’s important to know the difference between the Dijkstra and the BellmanFord algorithm.

|  |
| --- |
| void MatrixWDigraph::BellmanFord(const int n, const int v){      // Single source all destination shortest paths with negative edge lengths.      for(int i=0;i<n;i++){          dist[i]=length[v][i];      }      for(int k=2;k<=n-1;k++){          for(each u such that u!=v and u has at least one incoming edge){              for(each<i,u> in the graph){                  if(dist[u]>dist[i]+length[i][u])                      dist[u]=dist[i]+length[i][u]              }          }      }  } |

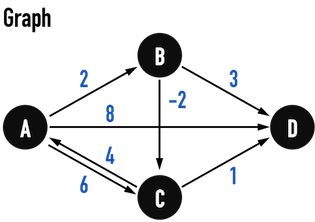
* Please explain the difference between the BellmanFord and the Dijkstra algorithm. Your answer should at least contain the difference of basic idea, and the time complexity.
* Please explain how can you apply the BellmanFord algorithm to find the shortest paths of the graph below. You may try to practice how to implement the algorithm.
  1. 

1. Although BellmanFord cannot find the shortest path of a graph with negative-cycle, it can be modified to check whether the graph has negative-cycles. Please explain how this can be done.
2. The Floyd-warshall algorithm is used to solve the All-Pairs Shortest Path problem. Please answer the following questions :

(1) What does All-Pairs Shortest Path mean?

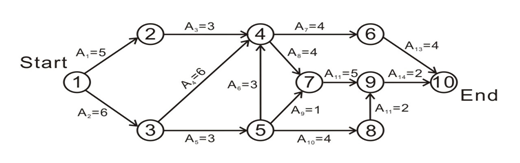
(2) Explain the concept of the Floyd-Warshall algorithm.

1. Please use Floyd-warshall algorithm to find the shortest path of all vertices in following graph.

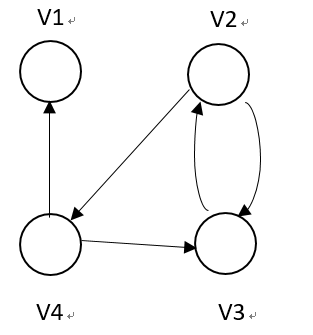


1. (1) Explain the concept of the AOE Network (Activity on Edge) and define what is a critical path.

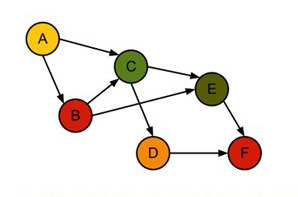
(2) What is the critical path of the AOE network below? Explain your method.



1. Please write a pseudo code that can implement your method to find a transitive closure of the graph below.



1. Please show step-by-step how you apply the DFS to find the topological order of the graph vertices.



1. Please give a practical example of applying the topological sorting or topological ordering. You must explain the meaning of the nodes and the edges in your example.