Computer Science 3600 (Winter 2024): Assignment #2 Supplementary Questions #2–5

2. (10 marks) Determine the optimal 0/1 knapsack-load for the set of items $U = \{1, 2, 3, 4, 5, 6\}$ and $B = 6$ using the algorithm given in Lecture #11 of the class notes. The sizes and values of the items in U are as follows:

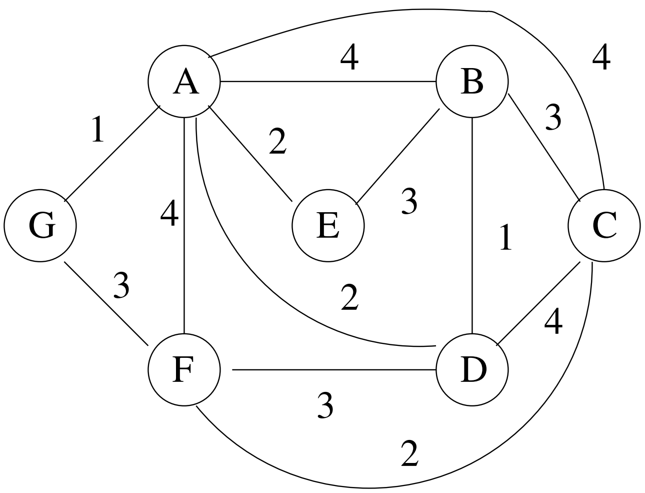
i	$s(i)$	$v(i)$
1	4	5
2	2	4
3	3	4
4	5	4
5	2	1
6	1	3

Show the filled-in dynamic programming value and backpointer matrices, the "backpointer path" in the backpointer matrix that gives an optimal knapsack-load, and the knapsack-load associated with that path.

3. (10 marks) For the problem below, give a pseudocode algorithm and a parameterized asymptotic worst-case time complexity for that algorithm. Note that this algorithm must run in parameterized polynomial time, *i.e.*, all terms excluding the variables denoting the time complexities of used operations must be polynomials in the input size.

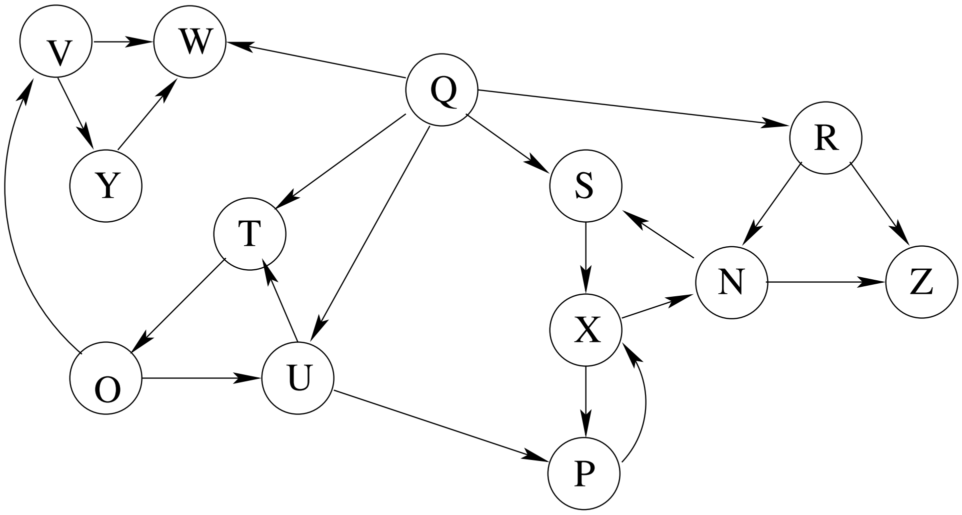
- Given a connected undirected graph $G = (V, E)$ and three non-overlapping vertex-subsets $S, I, F \subset V$, a **transit path** in G from $x \in S$ to $y \in F$ is a path in G from x to y that passes through at least one vertex in I. Given G, S, I, F, x, and y, compute the length of the shortest transit path (in terms of number of edges in the path) from x to y in G. You may use the following operations:
  - `SP(G, x, y)`: Returns the length of the shortest path in G between x and y.
  - `size(X)`: Returns number of vertices in vertex-set X.
  - `getVertex(X, i)`: Returns the ith vertex in vertex-set X, where $1 \leq i \leq size(X)$.

4. (14 marks) Consider the following weighted undirected graph:



- **a) (7 marks)** Show how Kruskal's minimum spanning tree algorithm works on this graph. Give the graph at the end of the execution of the algorithm on page 569 of the textbook, with all tree-edges and the order in which each tree-edge was added clearly marked.
- **b) (7 marks)** Show how Prim's minimum spanning tree algorithm works on this graph relative to root-vertex $r =$ G. Give the graph at the end of the execution of the algorithm on page 572 of the textbook, with all tree-edges and the order in which each tree-edge was added clearly marked.

5. (21 marks) Consider the following directed graph:



Assume that the algorithms below consider vertices in alphabetical order and that each adjacency list is ordered alphabetically.

a) (7 marks) Show how breadth-first search works on this graph. Give the graph at the end of the execution of the BFS algorithm on page 532 of the textbook when the search is started at vertex X, with the *d*- and π-values for all vertices as well as all BFS-search tree edges clearly marked.

b) (7 marks) Redo part (a) above with the BFS algorithm starting at vertex Q.