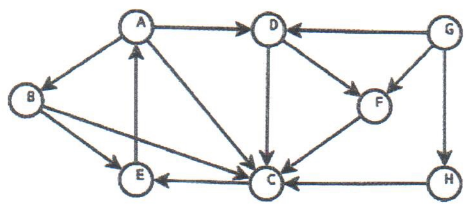
Algorithms
Comprehensive Exam
(Spring 2021)

SHORT QUESTIONS **(Answer all six questions, each carrying 7 points.)**

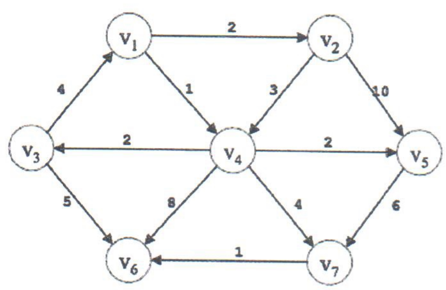
1. Derive a big-O (upper bound), a big-Omega (lower bound), and a big-Theta (tight bound) estimations, respectively for $f(n) = n^2 + 10^2 \cdot n \cdot logn + 10^3$, where n is a positive integer.

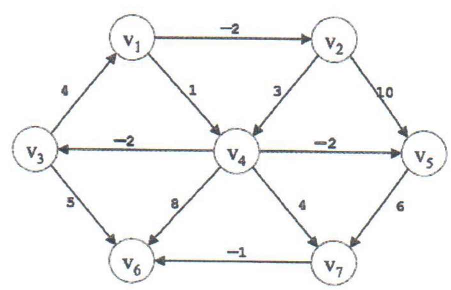
2. For an open-address hash table with load factor $\alpha = n/m < 1$, prove that
   (a) the expected number of probes in unsuccessful search under uniform hashing is at most $1/(1 - \alpha)$, and
   (b) the expected number of probes in a successful probe under uniform hashing being at most $(1/\alpha) \cdot \ln(1-\alpha)-1$? (Just give a proof sketch, explaining how many probes are needed to locate existing keys.)

3. BFS/DFS. Give the visited vertex order for each type of graph search, starting with vertex A.

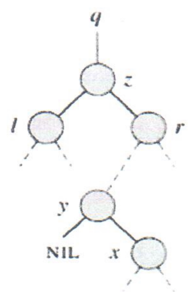


4. Shortest Path.
   (a) Run Dijkstra's algorithm on the directed graph shown below (on the left), starting at vertex V_1. Please show each step.
   (b) Run Bellman-Ford algorithm on the directed graph shown below (on the right), starting at vertex V_1. Please show each step.

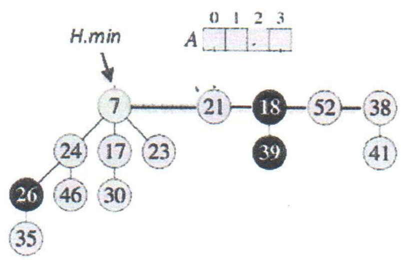




5. When the root of a binary subtree, z, is deleted, as depicted below, show the resulting subtree.



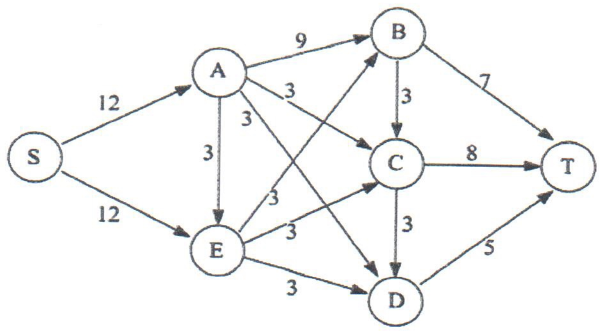
6. A Fibonacci min-heap relies on the procedure of CONSOLIDATE to merge trees in the root list upon the operation of extracting the minimum node. Show steps involved and the resulting heap after *H.min* is extracted from the Fibonacci min-heap given below.



LONG QUESTIONS (Answer all four questions, each carrying 15 points.)

1. Run the Ford-Fulkerson algorithm to compute the max flow of the flow network shown below. Please show the corresponding residual network at each step when running the Ford-Fulkerson algorithm.

   Derive the min-cut of the flow network and show each step.



2. Knapsack problem. Assume that we have a knapsack with a max weight capacity $W = 10$ pounds. Our objective is to fill the knapsack with items such that the total value is maximum. The following table contains the items along with their value and weight. Note that no duplicate one for each item exists.

Item i	1	2	3	4
Value ($)	10	30	40	50
Weight (W) (pounds)	3	6	4	3

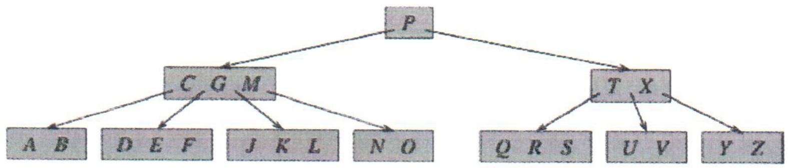
Please fill out the given table using a dynamic programming algorithm to find the maximum value of the knapsack.

For the last entry of the table, please explain how the algorithm computes the value.

Please give the exact items making the maximum value of the knapsack by checking the table, and explain.

[Weight] [Item]	0	1	2	3	4	5	6	7	8	9	10
0											
1											
2											
3											
4											

3. Given the initial B-tree with the minimum node degree of $t = 3$ below, show the results (a) after deleting the key of T, (b) followed by deleting the key of C, (c) then by inserting the key of J_2, with $J < J_2$, and (d) then by deleting the key of Y. (Show the result after each deletion and after each insertion.)



4. The NP-complete class contains a fraction of NP problems, which contain all P problems.
   (a) How do you prove the very first NP-complete problem?
   (b) The traveling-salesman problem of a complete undirected weighted graph is NP-complete, and it has a 2-approximation solution in polynomial time. Outline such an approximate solution and sketch a brief proof to demonstrate that such a solution satisfies 2-approximation.

Good Luck!