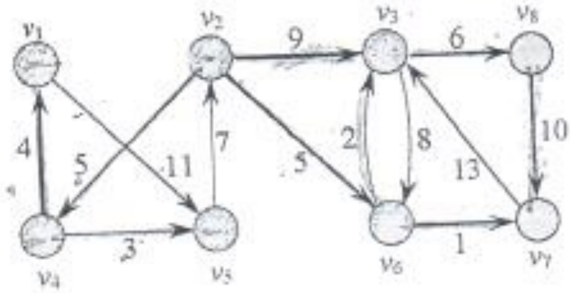
Fall 2019

CSCE 500 Final Exam

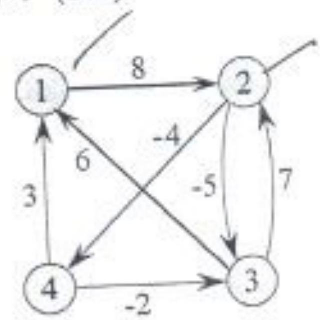
12/12/2019

1. Follow depth-first search (*DFS*), starting from Node v_1 to traverse in the following graph. Mark (1) the type of every edge and (2) the discovery and the finish times of each node. **(8%)**



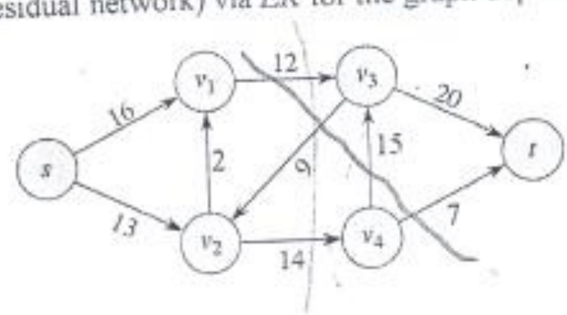
2. The Floyd-Warshall algorithm (*FW*) obtains all pairs of shortest paths in a weighted directed graph, with its pseudo code listed below. Consider the following graph, with its vertices labeled 1, 2, 3, and 4. Derive all distance matrices $D^{(k)}$, for $0 \le k \le 4$, following *FW* so that the $d_{ij}^{(n)}$ element of final matrix $D^{(n)}$ denotes $\delta(i, j)$ for every vertex pair $<i, j>$. **(8%)**

```
FLOYD-WARSHALL(W, n)
D^(0) = W
for k = 1 to n
    let D^(k) = (d_ij^(k)) be a new n × n matrix
    for i = 1 to n
        for j = 1 to n
            d_ij^(k) = min(d_ij^(k-1), d_ik^(k-1) + d_kj^(k-1))
return D^(n)
```



3. Prim's algorithm (*PM*) for minimum spanning trees (MSTs) follows greedy selection on nodes for expanding the tree progressively. Follow *PM* to establish an MST rooted at vertex v_1 over the graph given in Problem 1, with the edge of each expanding step (after one node addition) labeled in sequence, as $e_1, e_2, \ldots, e_5$. **(6%)**

4. The Edmonds-Karp algorithm (*EK*) follows the basic Ford-Fulkerson method with breadth-first search to choose the shortest augmenting path (in terms of the number of edges involved) for computing the maximum flow iteratively from vertex s to vertex t in a weighted directed graph. Illustrate the maximum flow computation process (including the augmenting path chosen in each iteration and its resulting residual network) via *EK* for the graph depicted below. **(10%)**



5. Solve the recurrence of $T(n) = 2 \cdot T(n/4) + 4 \cdot T(n/8) + c \cdot n$ by either a substitution or recursion-tree method. (7%)

Derive asymptotic upper and lower bounds for $T(n) = 2T(n/8) - n^{1/3}$ using the master theorem. (3%)

6. Given two hash functions of h_1 and h_2 for Cuckoo hashing under two tables, T_1 and T_2, briefly describe the steps involved in inserting a record with the key of K_{new}. (7%)

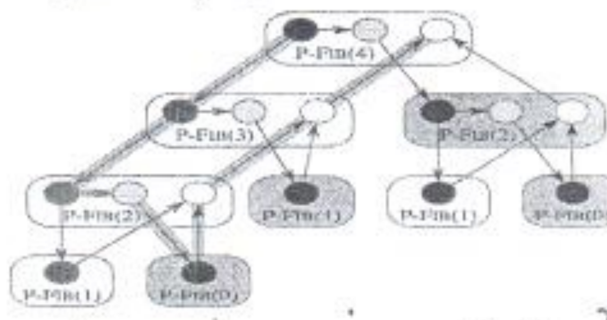
Under what condition(s) a new record cannot be inserted into a Cuckoo hashing table? (3%)

7. 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? (4%)
   (b) After NP-complete problems are proven, how do you show a new problem at hand to be NP-complete? (2%)

The traveling-salesman problem of a complete undirected weighted graph is NP-complete, and it has a 2-approximation solution in polynomial time given in the textbook.
   (a) Outline such an approximate solution. (4%)
   (b) Sketch a brief proof to demonstrate that such a solution satisfies 2-approximation. (8%)

8. A pseudo code for computing FIB(4) via $F_i = F_{i-1} + F_{i-2}$ without multithreading support and the DAG (directed acyclic graph) denoting its computation are shown below.
   (a) Write a code version with multithreading support. (4%)
   (b) Label (from 1 to 17) on the DAG under P_1 (with one thread); also label (from 1 to 8) on the DAG under P_∞ (with an infinite thread count). (6%)
   (c) What is the inherent parallelism expression in general? (2%)

```
FIB(n)
if n ≤ 1
    return n
else x = FIB(n − 1)
    y = FIB(n − 2)
    return x + y
```



$- 4 \times \frac{n}{32}$

9. Consider the matrix-chain multiplication problem for four matrices A_1, A_2, A_3, A_4, with their sizes being 30×10, 10×50, 50×40, and 40×20, respectively. Follow the tabular, bottom-up method in the procedure of MATRIX-CHAIN-ORDER below to construct tables that keep respectively entry $m[i, j]$ for all $1 \le i, j \le 4$ and entry $s[i, j]$ for $1 \le i \le 3$ and $2 \le j \le 4$ to get the optimal parenthesized multiplication result.
   (a) Construct the two tables, with their entry values shown. (16%)
   (b) Show the parenthesized multiplication of the matrix-chain. (2%)