Computer Science 3711 / Engineering 5891
Winter 2001

Final Exam

April 18, 2001

Instructor:
T. Wareham

NAME: ________________________ STUDENT ID #: ______________

- This exam will be 120 minutes long.
- This exam has 12 pages (including this cover page). There are 7 questions, most of which have multiple parts.
- Please answer all questions in the space provided on this exam; if you find it necessary to continue an answer on the back of a sheet of paper, that is fine, but please make a note on the front side, *e.g.*, "answer cont'd on back".

Question	Mark
1.	
2.	
3.	
4.	
5.	
6.	
7.	
TOTAL	

1. (8 marks) Derive an upper bound for $T(n)$ by the iteration method, where

$$T(n) = \begin{cases} 5 & n = 1 \\ 3T(n-1) + 2 & n > 1 \end{cases}$$

2. (12 marks) Circle the letters associated with the appropriate answers in the following multiple choice questions. Note that some of these questions may have more than one answer whose letter needs to be circled.

(a) (6 marks) Which of the following is true of the function $T(n) = 2^{n-3}$?

a) $T(n)$ is $\Omega(n^2)$
b) $T(n)$ is $O(n^{100})$
c) $T(n)$ is $O(n^n)$
d) $T(n)$ is $\Theta(2^{n+2})$

(b) (6 marks) Which of the following is true of the function $T(n) = n^{\log n}$?

a) $T(n)$ is $\Omega(n^{1000})$
b) $T(n)$ is $O(2^n)$
c) $T(n)$ is $O(n^{100})$
d) $T(n)$ is $\Theta(n^{(\log n)+3})$

3. (9 marks)

Consider the following algorithm that operates on a graph $G = (V, E)$:

```
sum = 0;
FOR all v in V DO
    k = T1(G,v);
    IF (degree(v) > 4) THEN
        for all e in E - {(u',v') | u' = v or v' = v}
                     such that T2(G,e) = TRUE DO
            l = T3(G,v,e);
            sum = sum + (k * l);
```

Assume that `degree(v)` can be computed in $O(1)$ time. Give the time complexity of the algorithm above when the T1, T2, and T3 operations require

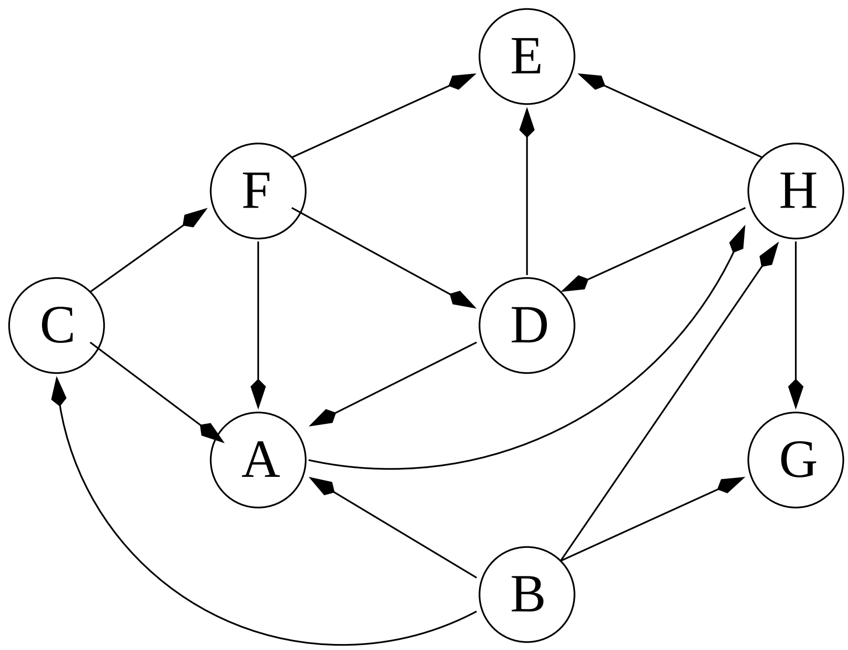
i) $O(1)$, $O(1)$, and $O(1)$ time, respectively

ii) $O(1)$, $O(1)$, and $O(\log |E|)$ time, respectively; and

iii) $O(|E|^3)$, $O(\log |E|)$, and $O(1)$ time, respectively.

4. (34 marks)

a) (8 marks) Consider the following directed graph:



Give the graph at the end of the execution of the DFS algorithm, with the d- and f-values of all vertices as well as the types of all edges clearly marked. Assume that the algorithm considers vertices in alphabetical order and that each adjacency list is ordered alphabetically.

b) (10 marks) Run Dijkstra's algorithm on the directed graph in Table 1 (see page 5) using vertex x as the source vertex. Show the d and π values and the vertices in set S after each iteration of the **while** loop by filling in the appropriate values on the spare copies of the graph given in this table. Did Dijkstra's algorithm correctly compute the shortest-path distances from x to every other vertex in this graph? If not, why not?

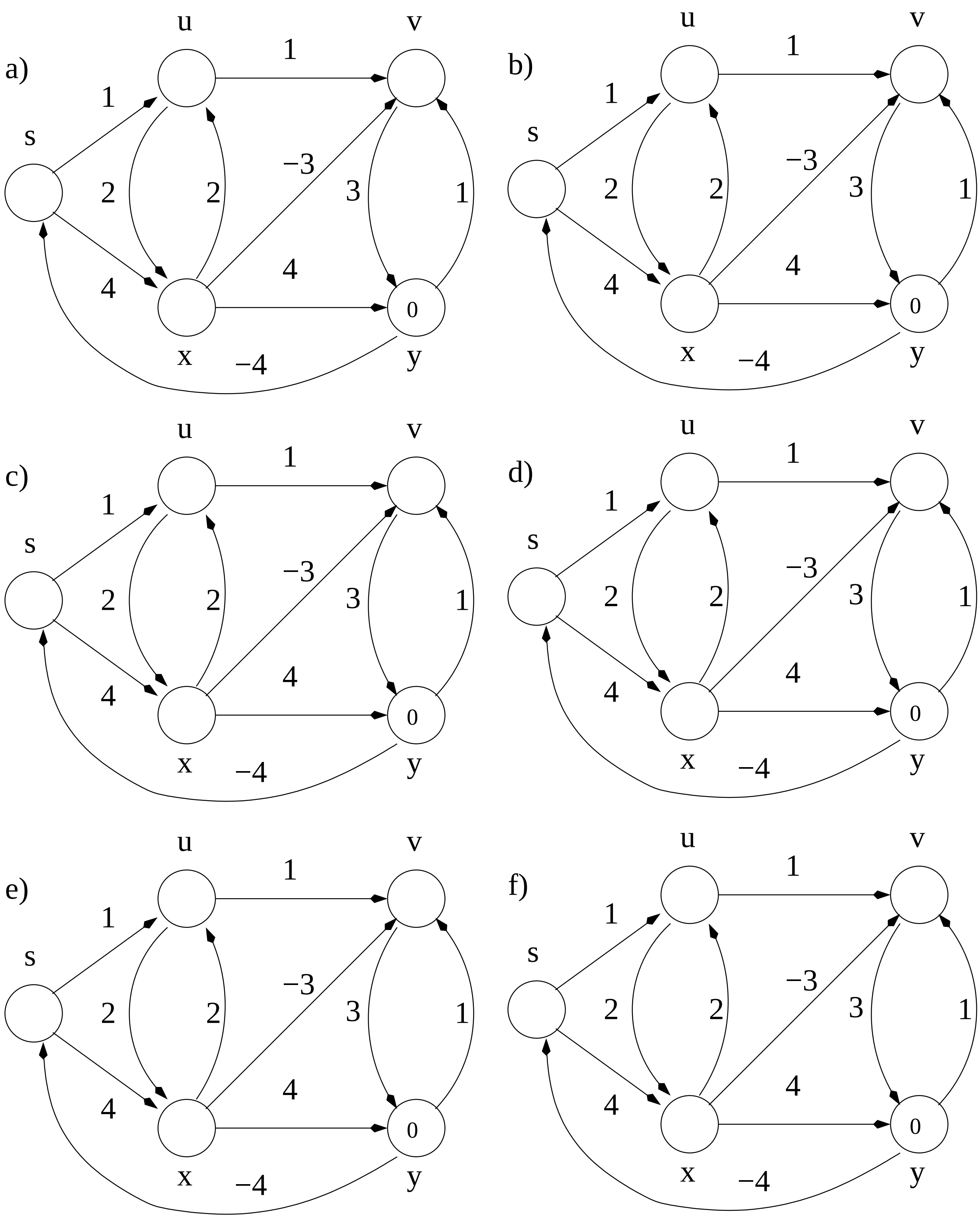


Table 1: Answer for Question #4(b).

c) (16 marks) Each of the problems described in this part can be solved using one or more calls to one of the graph algorithms we have examined in this course. For each problem, state the graph algorithm that you will be calling as a subroutine and give the algorithm for the given problem that makes these calls.

i) (8 marks) A cable television company has developed a network consisting of a set C of two-way fibre optic cables linking a set R of local rebroadcast stations. Each cable has a different positive bandwidth and the maximum bandwidth over all cables is B. The company engineers need a high-bandwidth multicast scenario to use in troubleshooting the network, *i.e.*, they need to find a spanning tree for the stations in the network such that the sum of the bandwidths of all cables in the spanning tree is the largest possible.

ii) (8 marks) The cable television company in part (i) has also decided to place specialized channel-subscription equipment at each station in a subset of its rebroadcast stations. To evaluate the cost of the placement scheme, they need to be able to find, for every rebroadcast station r *not* getting the equipment, the shortest number of fibre-optic cables linking r to any station getting the equipment.

5. (20 marks)

a) (10 marks) Determine the longest common subsequence (LCS) of the strings **GUGCA** and **GCA** using the dynamic programming algorithm **as given in the textbook**. Fill in the dynamic programming matrix given below, show all matrix-cell backpointers, show one of the backpointer paths that gives an optimal LCS, and show the LCS corresponding to this path.

			G	U	G	C	A
		0	1	2	3	4	5
	0						
G	1						
A	2						
C	3						

b) (6 marks) Given a pair of strings x and y over a common alphabet Σ, consider the problem of finding the longest string over Σ that is a substring of x and a subsequence of y. For example, given $x =$ **CGC** and $y =$ **CUGAG**, the longest such string would be **CG**. Give the modification of the recurrence underlying the dynamic programming algorithm for the longest common subsequence problem that solves this new problem.

c) (4 marks) Give an asymptotic worst-case upper bound on the space used by your algorithm in part (b). Is this bound the same as that for the algorithm you used in part (a)? Does your algorithm in part (b) actually use less space than the algorithm you used in part (a)? If so, how much less?

6. (13 marks)

a) (10 marks) What is the optimal Huffman code for the following set of frequencies?

a:29 b:12 c:25 d:15 e:19

Please show the priority-queue of code-trees at each stage in the execution of Huffman's algorithm, and give the code-word associated with each symbol above under the code derived from the Huffman tree.

b) (3 marks) What is the difference between problems that have dynamic programming algorithms and problems that have greedy algorithms?

7. (24 marks)

Circle the letters associated with the appropriate answers in the the multiple choice questions in parts (a–c). Note that some of these questions may have more than one answer whose letter needs to be circled.

Suppose we have four decision problems A, B, C, and D such that $B \leq_p A$, $C \leq_p A$, $C \leq_p B$, and $D \leq_p C$, where $X \leq_p Y$ means that there is a polynomial-time (many-one) reduction from X to Y, *i.e.*, X reduces to Y.

(a) (5 marks) What is implied if we know that B is NP-hard?

a) B is not solvable in polynomial time.
b) if B is solvable in polynomial time then $P = NP$.
c) C is NP-hard.
d) A is NP-hard.
e) A is not solvable in polynomial time unless $P = NP$.

(b) (5 marks) What is implied if we know that C is solvable in polynomial time and A is NP-hard?

a) D is solvable in polynomial time.
b) B is solvable in polynomial time.
c) $P \neq NP$.
d) B is not solvable in polynomial time unless $P = NP$.
e) $P = NP$.

(c) (5 marks) What is implied if we know that A is solvable in polynomial time and C is NP-hard?

a) D is solvable in polynomial time.
b) $P = NP$.
c) B is solvable in polynomial time.
d) $P \neq NP$.
e) A is NP-hard.

Consider the following two problems:

CLIQUE-I
INPUT: An undirected graph $G = (V, E)$ and an integer k.
QUESTION: Is there a clique in G of size at least k, *i.e.*, is there a subset $V' \subseteq V$, $|V| \geq k$, such that for all $u, v \in V'$, $(u, v) \in E$?

CLIQUE-II
INPUT: An undirected graph $G = (V, E)$.
OUTPUT: One of the largest cliques in G.

(d) (3 marks) Is CLIQUE-I in the class NP? If so, why? If not, why not?

(e) (3 marks) Is CLIQUE-II in the class NP? If so, why? If not, why not?

(e) (3 marks) Give a polynomial-time reduction from CLIQUE-I to CLIQUE-II, *i.e.*, show that CLIQUE-I reduces to CLIQUE-II.