

Computer Science 3711
Winter 2005

Final Exam

April 13, 2005

Instructor:
T. Wareham

NAME: _____

STUDENT ID #: _____

- This exam will be 120 minutes long.
- This exam has 9 pages (including this cover page). There are 5 questions, all of which have multiple parts.
- 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.	14
2.	26
3.	24
4.	36
5.	20
	120

1. (14 marks)

a) (8 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.

i) (4 marks) Which of the following is true of the function $T(n) = n^2 2^{2 \log_2 n}$?

- a) $T(n)$ is $O(n 2^{\log_2 n^6})$ b) $T(n)$ is $\Omega(2^{3 \log_2 n})$
c) $T(n)$ is $\Omega(n^5)$ d) $T(n)$ is $\Theta(2^{30} n^4)$

ii) (4 marks) Which of the following is true of the function $T(G) = |V| |E| \log_2 |E|^4$ relative to undirected graph $G = (V, E)$?

- a) $T(G)$ is $\Theta(|V|^2 \log_2 |V|^4)$ for complete graphs
b) $T(G)$ is $\Omega(|E|^2 \log_2 |V|^6)$ for complete graphs
c) $T(G)$ is $O(|V|^2 \log_2 |V|^2)$ for acyclic graphs
d) $T(G)$ is $\Omega(|E|^3 \log_2 |E|)$ for acyclic graphs

b) (6 marks) Give a recurrence for the worst-case time complexity of the following recursive algorithm:

```
procedure FUNKY-REC(m, n)
  mult = m
  if (n < 3)
    return(m)
  else if (n > 10)
    for (i = 1; i <= n * log2(m); i++)
      for (j = 1; j <= n; j++)
        mult = FUNKY-ITR(i * j) * FUNKY-REC(m, n/4)
  else if (n < 20)
    for i = 1 to log2(m) do
      mult = FUNKY-ITR(i * i * m) * FUNKY-REC(m, n/5)
  else if (n > 5)
    for i = 1 to n * m do
      mult = FUNKY-ITR(m) * FUNKY-REC(m, n/2)
  else
    for (i = 1; i <= n * n * n * n; i++)
      for (j = 1; j <= m; j++)
        mult = mult * FUNKY-ITR(i) * FUNKY-REC(m, n/3)
  return(mult)
```

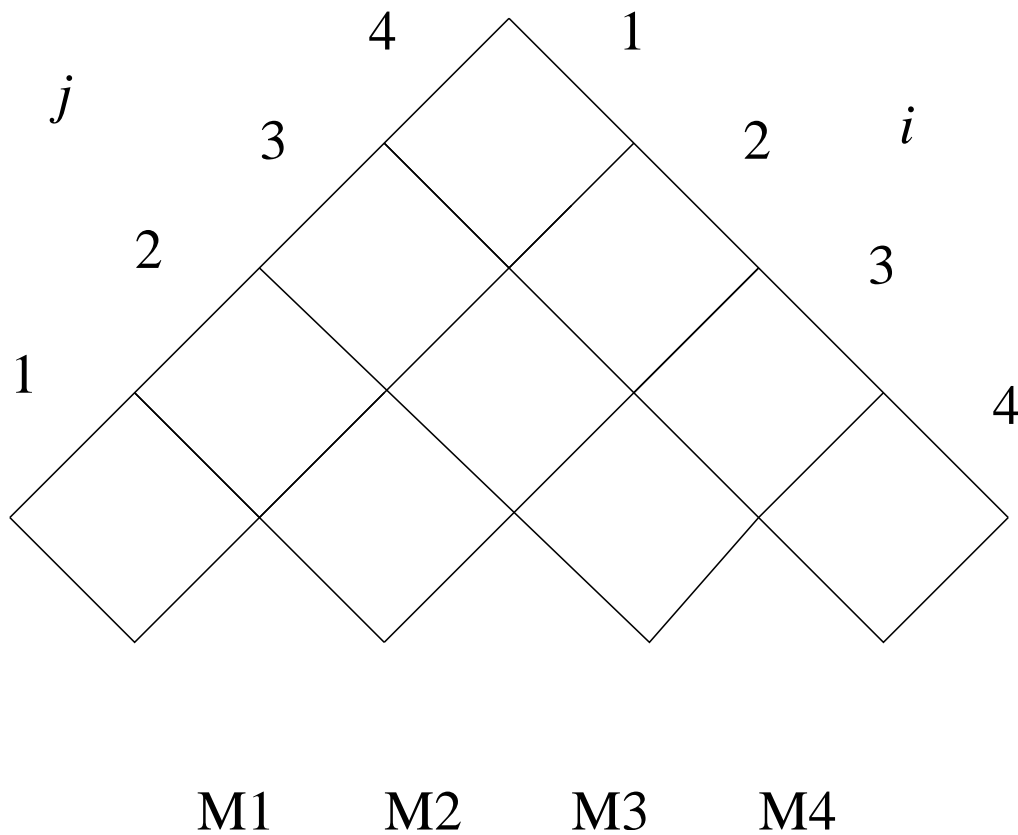
Assume that procedure FUNKY-ITR(x) runs in $O(x^3)$ time.

2. (26 marks)

- a) (14 marks) Consider the the dynamic programming algorithm for the Matrix Chain Parenthesization problem defined by the recurrence

$$m[i, j] = \begin{cases} 0 & i = j \\ \min_{i \leq k < j} m[i, k] + m[k + 1, j] + p_{i-1}p_jp_k & i < j \end{cases}$$

where the input is a matrix-chain dimension list $p = \langle p_0, p_1, \dots, p_n \rangle$ specifying a matrix-chain $M_1M_2 \dots M_n$ with dimensions $(p_0 \times p_1), (p_1 \times p_2), \dots, (p_{n-1} \times p_n)$. Given matrix-chain dimension list $p = \langle 2, 3, 4, 2, 5 \rangle$, compute and output an optimal parenthesization of the corresponding matrix-chain relative to the recurrence above (that is, fill in the given dynamic programming table (including one backpointer per table-cell), show the backpointer “path” that gives an optimal parenthesization, and give the parenthesization associated with that backpointer-“path”).



b) (12 marks) Consider the following algorithm:

```
DFS-V(i, sol, U, BL, BU, val)
  if (SIZE(sol) > BU)
    print("pruned: solution too big for knapsack")
  else if (i == n)
    if (SIZE(sol) < BL)
      print("pruned: solution too small for knapsack")
    else if (VALUE(sol) < val)
      print("pruned: solution too cheap for knapsack")
    else
      print(sol)
  else
    DFS-V(i + 1, sol, U, BL, BU, val)
    DFS-V(i + 1, UNION-COPY(sol, U[i]), U, BL, BU, val)
```

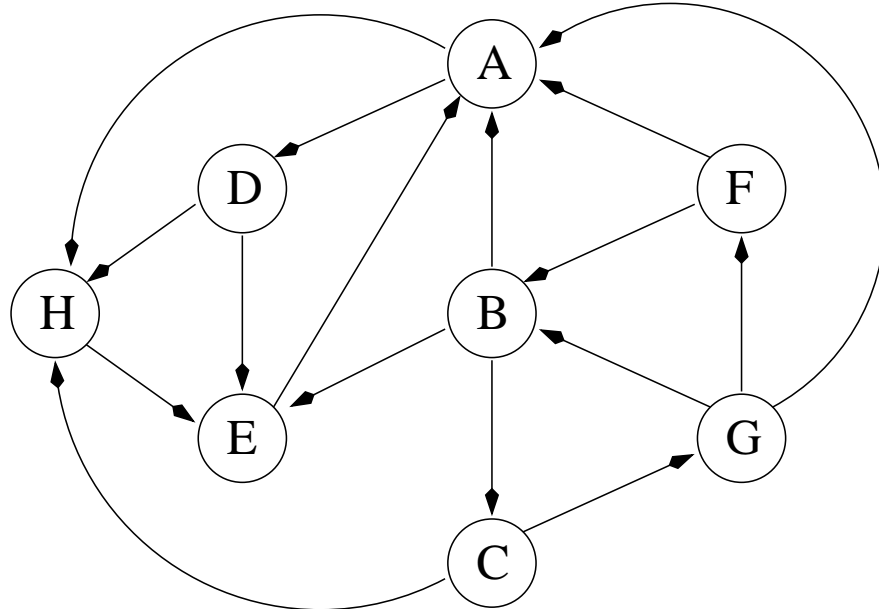
The problem solved here is a variant of 0/1 KNAPSACK that prints all knapsack-loads of value at least val whose sizes are between BU and BL inclusive. In this algorithm, U is the set of items, n is the number of items in U , BL and BU are the lower and upper knapsack size bounds, sol is a subset of U , $SIZE(sol)$ returns the sum of the sizes of the items in sol , $VALUE(sol)$ returns the sum of the values of the items in sol , and $UNION-COPY(sol, U[i])$ returns a copy of sol to which the i th item in U has been added. To answer this question, sketch the implicit tree of solution-nodes generated by the algorithm above (marking leaf-nodes which are printed by a circle and nodes that are pruned with a square) with the call $DFS-V(0, sol, U, BL, BU, val)$ when $U = \{X, Y, Z\}$ such that $size(X) = 2$, $size(Y) = 3$, $size(Z) = 2$, $value(X) = 1$, $value(Y) = 2$, and $value(Z) = 2$, sol is the empty set, and BL , BU , and val have the following values:

i) (6 marks) $BL = 2$, $BU = 4$, and $val = 2$:

ii) (6 marks) $BL = 4$, $BU = 7$, and $val = 4$:

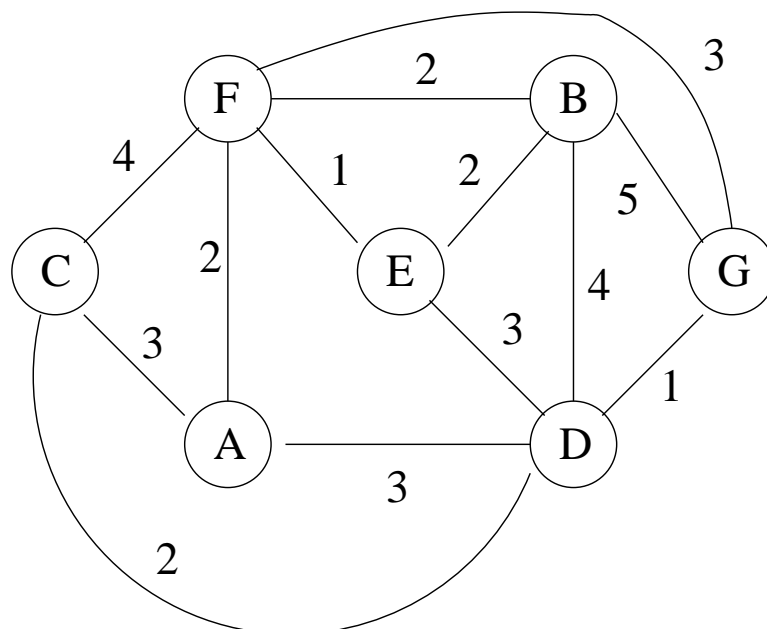
3. (24 marks)

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

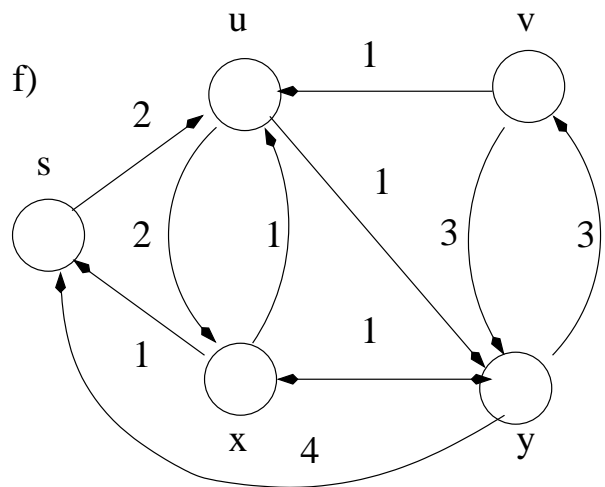
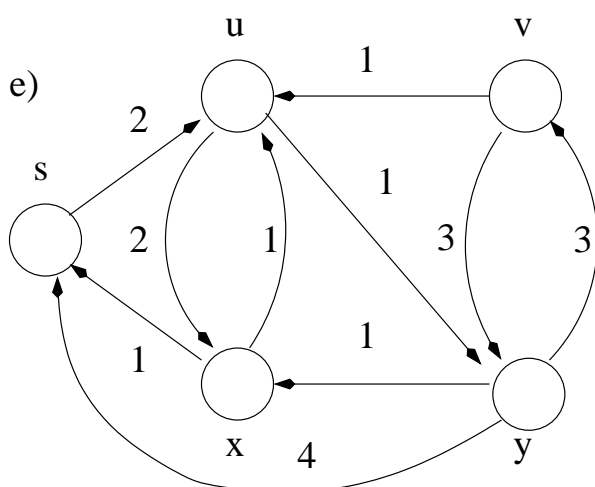
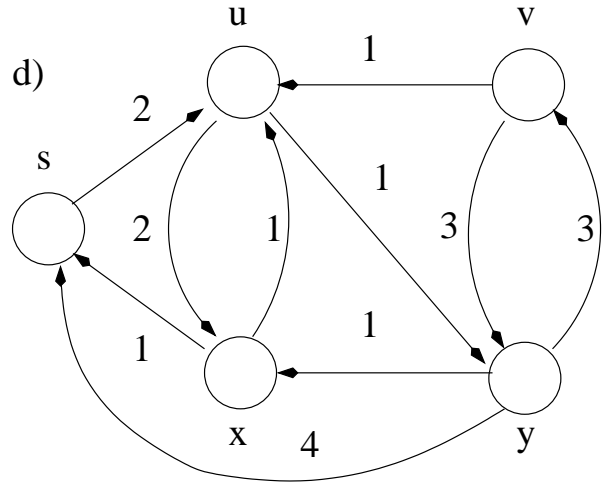
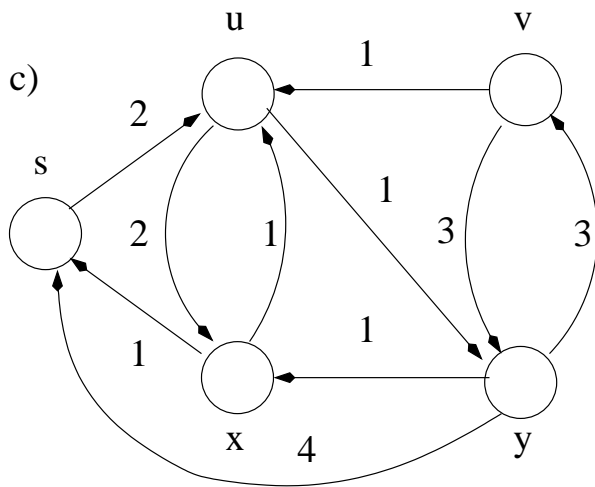
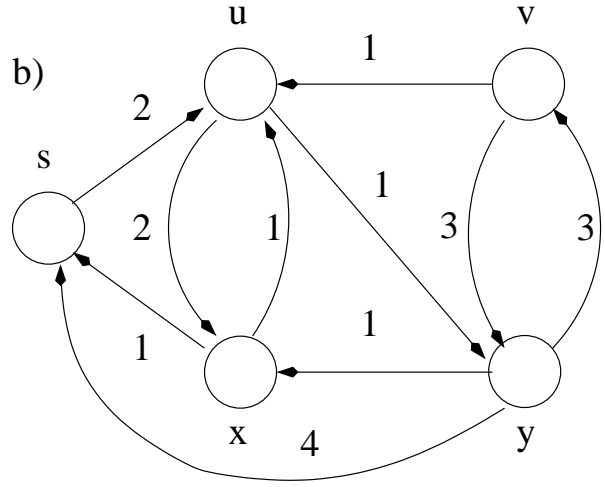
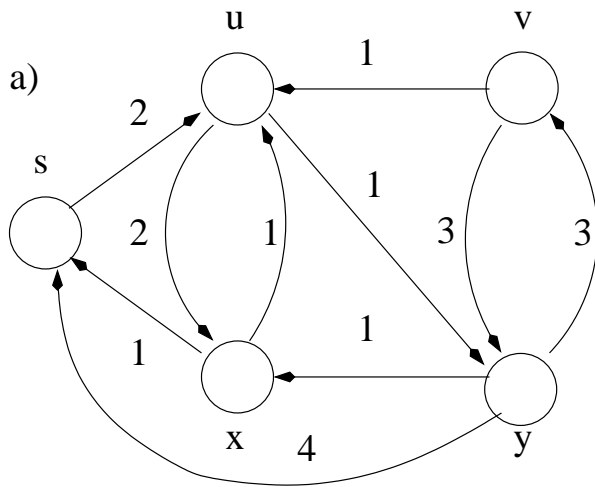


Give the graph at the end of the execution of the DFS algorithm, with the d - and f -values of all vertices as well as the types of all edges clearly marked. Assume that the algorithm considers vertices in alphabetical order and that each adjacency list is ordered alphabetically.

b) (6 marks) Show how Prim's minimum spanning tree algorithm works on the undirected edge-weighted graph below relative to root-vertex $r = B$. Mark all tree-edges and the order in which each tree-edge was added.



c) (10 marks) Run Dijkstra's algorithm on the directed graph below 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.



4. (36 marks)

- a) (18 marks) Consider the following problem: You have been asked to help schedule the stunts in movie scripts for a major Hollywood studio. Each movie script consists of a sequence of scenes, where each scene has an associated set-type (Urban / Country / Outer Space / Underwater) and a sequence of stunt-activities (Fall / Crash / Explosion). For example, Scene 3 may take place in Outer Space (O for short) and has the stunt-sequence Fall / Fall / Crash / Explosion / Explosion (or FFCEE for short). The studio lot consists of a group of movie sets of various types, each of which can support a particular sequence of stunts, and two-way roads connecting these sets together. A scene in a movie script can be scheduled for a particular set if the scene and the set are of the same type and the stunt-sequence of the scene is a subsequence of the stunt-sequence for that set. A movie script of n scenes is viable if there is a path of n road-connected sets on the studio lot such that scene i can be scheduled for set i on this path for $1 \leq i \leq n$.

Give pseudocode for an algorithm that solves the scheduling problem above – namely, given a scene-sequence $S = \langle (t_1, s_1), (t_2, s_2), \dots, (t_n, s_n) \rangle$ of set-type / stunt-sequence pairs such that $t_i \in \{U, C, O, U\}$ and s_i is a string over the alphabet $\{F, C, E\}$ and an undirected graph $L = (V, E)$ where each vertex v has an associated set-type $t(v)$ and stunt-sequence $s(v)$, determine whether or not there is a path $\langle v_1, v_2, \dots, v_n \rangle$ of n vertices in L such that $t_i = t(v_i)$ and s_i is a subsequence of $s(v_i)$ for $1 \leq i \leq n$ and $(v_i, v_{i+1}) \in E$ for $1 \leq i \leq (n-1)$. You may assume that no two adjacent scenes in scene-sequence S have the same set-type. Please give the asymptotic worst-case time complexity of your algorithm.

- b) (18 marks)** Let $G = (V, E, l)$ be an edge-labelled undirected graph such that each edge e has label $l(e) \in \{s, d\}$. Give pseudocode for a polynomial-time algorithm that computes a subset $E' \subseteq E$ of edges in G (if such an E' exists) such that in graph $G' = (V, E')$, there is a path between each pair of vertices in V and the number of edges with label d in E' is the smallest possible. Please give the asymptotic worst-case time complexity of your algorithm.

5. (20 marks) Circle the letters associated with the appropriate answers in the the multiple choice questions in parts (i–v). Note that some of these questions may have more than one answer whose letter needs to be circled.

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

i) (4 marks) What is implied if we know that B is solvable in polynomial time?

- a) G is not solvable in polynomial time.
- b) if G is solvable in polynomial time then $P = NP$.
- c) C is solvable in exponential time.
- d) F is solvable in polynomial time.

ii) (4 marks) What is implied if we know that A is NP -complete?

- a) G is NP -hard.
- b) B is NP -complete.
- c) D is NP -hard.
- d) C is not solvable in polynomial time.

iii) (4 marks) What is implied if we know that C is NP -hard and B is solvable in polynomial time?

- a) D may be solvable in polynomial time.
- b) $P = NP$.
- c) G is solvable in polynomial time.
- d) E is NP -complete.

iv) (4 marks) What is implied if we know that D is NP -hard and C is solvable in polynomial time?

- a) F is solvable in polynomial time.
- b) $P \neq NP$.
- c) A is solvable in exponential time.
- d) B may be solvable in polynomial time.

v) (4 marks) What is implied if we know that B is NP -hard, C is solvable in polynomial time, and F is in class NP ?

- a) F is solvable in polynomial time.
- b) $P = NP$.
- c) D is solvable in exponential time.
- d) C may be solvable in polynomial time.