CMPUT 204 Section B3
Prof. Culberson
FINAL April 20, 2000
CLOSED BOOK. NO Notes or Calculators.
Time 2 hours.
Answer all questions in space provided
Do scratch work on page backs



Last Name: ____________________

First Name: ____________________

Seat Number (on top of your desk) : ____________________

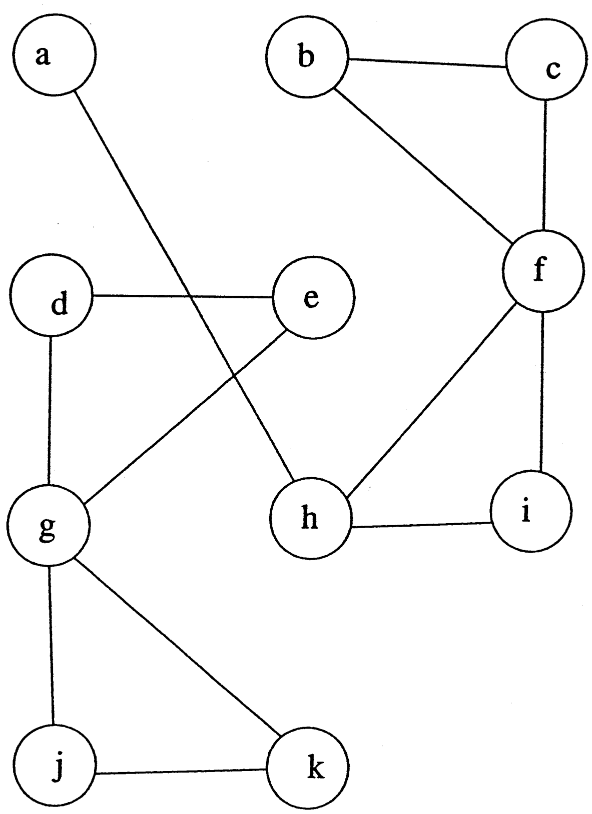
Make sure your name and ID is on the top of each page.

**Question 1** *Marks 4* Given a graph with $n$ vertices and $m$ edges where $m \in \Theta(n)$, how long does it take to determine the degree of every vertex, assuming that the graph is represented by (i) adjacency lists (ii) an adjacency matrix? Justify your answers.

(i)

(ii)

**Question 2** *Marks 4* For the following graph, list all articulation points and the bicomponents of the graph.



**Question 3** *Marks 2* Here is a description of the prime number decision problem.
Problem instance: "Is integer $n$ a prime?"
What is wrong with the following 'proof' that this problem is in NP?
Certificate: A set of integers $I = \{2, 3, \ldots, n-1\}$.
Verification: For each $x$ in $I$, check in $O(1)$ time to see that $x$ does not divide $n$ without a remainder.
Total verification time is $O(n)$.

**Question 4** *Marks 3* Put the following in order of their complexity, smallest to largest, clearly indicating when two functions are of the same order of complexity.

$$2^n \quad n^2 \quad n^2 + \log_2 n \quad n^2 \log_2 n \quad 2^{n \log_2 n} \quad n^2(\log_2 n)^2$$

**Question 5** *Marks 3* For the following algorithm, write a recurrence describing the worst case number of comparisons of X to an array element. Indicate clearly how the terms of your recurrence relate to the program. DO NOT try to solve this recurrence or explain the algorithm.

```
Proc TernaryS(A,first,last, int X){                    //ANSWER HERE
    if ( first >= last ) {
        if (A[first] == X) return yes
        else return no;
    }
    third = (last -first)/3;
    p = first + third;
    q = p+1+third;
    if (X <= A[p]) then
        return TernaryS(A,first,p,X);
     else if ( X <= A[q] ) then
        return TernaryS(A,p+1,q,X);
     else
        return TernaryS(A,q+1,last,X);
}
Initially called by TernaryS(A,1,n,X);
```

**Question 6** *Marks 2* Let $T(1) = 3000$ and $T(n) = 16T(n/4) + n^2$ for $n > 1$ and $n$ a power of 4 (that is, $n = 4^k$ for integer $k$). Give the simplest big-$\Theta$ expression for $T(n)$.

**Question 7** *Marks 10* Each line below describes one neighbor list of the adjancy list representation of an undirected weighted graph $G = (V, E, W)$, where the numbers are the weights on the edges, and the vertices are {a, ..., i}. Feel free to draw the graph.

| | | | | |
|---|---|---|---|---|
| a : | (b,10) | (c,5) | | |
| b : | (a,10) | (d,4) | (f,1) | |
| c : | (a,5) | (d,2) | (e,3) | |
| d : | (b,4) | (c,2) | (e,1) | (g,8) |
| e : | (c,3) | (d,1) | | |
| f : | (b,1) | (g,6) | | |
| g : | (d,8) | (f,6) | (h,1) | (i,1) |
| h : | (g,1) | (i,1) | | |
| i: | (g,1) | (h,1) | | |

7.a List the edges of a minimum spanning tree in the order found by the Prim-Dijkstra algorithm starting at vertex 'a'. (This is the algorithm that adds edges to an existing tree from a fringe list.)

7.b How many distinct MST's are there for this graph? Give your argument indicating how you arrived at this number.

**Question 8** *Marks 10* Let $A_1, \ldots, A_4$ be matrices with dimensions $2 \times 5$, $5 \times 3$, $3 \times 1$, $1 \times 4$ respectively. Recall that for $x \leq y$, $M(x, y)$ is the minimum number of scalar multiplications needed to compute $A_x \times A_{x+1} \times \ldots \times A_y$. By filling in the entries $M(x, y)$ in the table below, determine the minimum number of scalar multiplications needed to compute $A_1 \times \ldots \times A_4$.

M(x,y)

| x \ y | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |

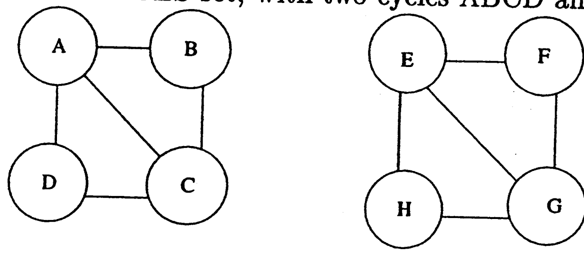
**Question 9** *Marks 10*

9.a Define a component of a graph.

9.b Give an efficient algorithm to solve the following problem.
Input: a graph $G = (V, E)$ with weights $w_1, \ldots, w_n$ on the vertices.
Goal: find the total weight of the vertices in each component in the graph

9.c Analyze the worst case running time of your algorithm as a function of the number of vertices $n$ and edges $m$.

**Question 10** *Marks 12* 2-cycles Problem:
Input: A Graph $G = (V, E)$.
Query: Does there exist a set of exactly two disjoint simple cycles in $G$ such that every vertex appears exactly once in exactly one of the cycles?
Here is an example graph from the YES set, with two cycles ABCD and EFGH.




Show that the 2-cycles problem is NP-complete. You may use the fact that Hamiltonian Cycle is NP-complete. (A Hamiltonian cycle in an undirected graph is a simple cycle that passes through every vertex exactly once.)

Hint: State all the steps required to show a problem is NP-complete. The majority of the marks is for clearly indicating how to show a problem is NP-complete. To obtain full marks, you will need a simple reduction, which is strongly hinted at in the example.

**Question 11** *Marks 10* Assuming that $1 \leq a \leq b$, the following program takes time exponential in $b - a$.

11.a Assume that the inputs are bounded with $1 \leq a \leq b \leq n$ for some integer $n$. Using the technique of memo-ization (i.e. using an extra memory array) modify this algorithm to run in time polynomial in $n$.

```
int hard(int x, int y)                                  // SHOW MODIFICATIONS HERE
{
        int w;
        if (y <= x) {return x;}
        else {


                w = hard(x,y-1) + hard(x+1,y);


                return w;
        }
}
int main()
{
        int a,b;


        scanf("%d %d",&a,&b);
        printf("%d\n",hard(a,b));
}
```

11.b Analyze the running time of your modified algorithm. Note: you are *not* required to analyze the original algorithm.

| Question | Mark | Out Of |
|---|---|---|
| 1 | ____ | 4 |
| 2 | ____ | 4 |
| 3 | ____ | 2 |
| 4 | ____ | 3 |
| 5 | ____ | 3 |
| 6 | ____ | 2 |
| 7 | ____ | 10 |
| 8 | ____ | 10 |
| 9 | ____ | 10 |
| 10 | ____ | 12 |
| 11 | ____ | 10 |
| Total | ____ | 70 |