University of Utah
School of Computing

CS 4150 **Midterm Exam** **March 2, 2017**

Solution

The average grade on the midterm was 80.

If you believe that a mistake was made in grading your exam, describe the problem on the cover sheet and return your exam to me by March 21. Do not modify any of your answers.

Master Theorem

If $T(n) = aT(n/b) + O(n^d)$ for constants $a > 0$, $b > 1$, $d \geq 0$, then

$$T(n) = \begin{array}{ll} O(n^d) & \text{if } d > \log_b a \\ O(n^d \log n) & \text{if } d = \log_b a \\ O(n^{\log_b a}) & \text{if } d < \log_b a \end{array}$$

1. The worst cases of four algorithms are carefully timed on dense graphs with n vertices (the Bellman-Ford algorithm), binary heaps of size n (DeleteMin), sparse directed graphs with n vertices (the strongly-connected components algorithm), and n-bit integers (Karatsuba multiplication). A different computer is used to time each implementation.

The results of the timing experiments are summarized below. All times are expressed in the same (unspecified) units.

n	Algorithm W	Algorithm X	Algorithm Y	Algorithm Z
2000	1	50	100	4
4000	8	60	300	8
8000	64	70	900	16
16000	512	80	2700	32
32000	4096	90	8100	64

Identify the algorithms by writing W, X, Y, and Z in the appropriate boxes below.

Algorithm	Identity
Bellman-Ford	W
DeleteMin	X
SCC Algorithm	Z
Karatsuba Multiplication	Y

2. In each of the modular equations below, solve for x. In each case, x must be a non-negative integer less than the modulus.

N

Expression	Value of x
$(x + 3) \equiv 1 \pmod{7}$	5
$(5 * 4) \equiv x \pmod{7}$	6
$9876543210^{783748973489738378378378} \equiv x \pmod{9876543211}$	1
$7^{10293987756798768401} \equiv x \pmod{48}$	7

3. Consider the following method:

```
    public static int f(int n) {
        int total = 0;
        for (int i = 0; i < n; i += 3) {
            for (int j = i; j < n; j++) {
                total++;
            }
        }
        return total;
    }
}
```

What does `f(3*k)` return? Assume that $k > 0$. Give your answer in terms of k. *Give a simplified expression that does not contain a summation.*

Answer	$\frac{3}{2}k(k+1)$

4. For each pair of functions below, write

- $\Theta(g)$ if f is $\Theta(g)$

Otherwise, write

- $O(g)$ if f is $O(g)$ or
- $\Omega(g)$ if f is $\Omega(g)$

f	g	Answer
$2n^3$	$3n^2$	$\Omega(g)$
$\log(n!)$	$n \log n$	$\Theta(g)$
$n \log n$	$n \log(n^2)$	$\Theta(g)$
3^n	$8^{n/2}$	$\Omega(g)$

5. Use the Master Theorem to derive a tight upper bound on the following two recurrences:

$$\begin{aligned} T_1(n) &= 8\,T_1(n/7) + O(n) \\ T_2(n) &= 4\,T_2(n/2) + O(n^2) \end{aligned}$$

Recurrence	Bound
T_1	$O(n^{log_7 8})$
T_2	$n^2 \log n$

6. Suppose you have a min binary heap containing the integers 1 through 15. The levels of the heap are numbered from 1 to n, where the root is the first level and there are n levels in all. Answer the following question about this binary heap.

Question	Answer
What number is stored at the root?	1
What is n?	4
What is the smallest number that could possibly be in the bottom level?	4
What is the largest number that could possibly be in level 2?	9

7. Indicate whether each of the following is a best practice for making timing measurements by writing "yes" or "no" in its box.

Advice	Yes or no?
Don't measure small intervals	yes
Account for timing overhead	yes
Avoid cold starts	yes
Time nothing that uses the heap	no

8. Each of the following algorithms has a $\Theta(n)$ worst case. Give the average case complexity of each.

Algorithm	Average case
Lookup in an n-element hash table	$\Theta(1)$
Lookup in an n-element binary search tree	$\Theta(\log n)$
Partitioning an n-element array around a pivot	$\Theta(n)$
Adding to the end of an n-element dynamic array	$\Theta(1)$

9. Use $O()$ and $\Omega()$ notation to give the tightest possible upper and lower bounds on the time required to determine whether there is a path between every pair of vertices in an undirected graph with V vertices and E edges.

Do this once assuming that the graph is represented as an adjacency matrix and again assuming that the graph is represented as an adjacency list.

	Matrix	List
Upper bound on worst case	$O(V^2)$	$O(E+V)$
Lower bound on best case	$\Omega(V)$	$\Omega(1)$

10. Let G be a directed acyclic graph on which a depth-first search has been performed. Classify the following statements as true or false.

Statement	True or False?
The vertex with the smallest pre time must be a source	false
The vertex with the largest pre time must be a source	false
The vertex with the smallest post time must be a sink	true
The vertex with the largest post time must be a sink	false

11. Consider the following four functions:

- $f_1(n) = \log(n^2)$
- $f_2(n) = n^{2/3}$
- $f_3(n) = n$
- $f_4(n) = \sqrt{n} \log n$

Put these functions in order of increasing asymptotic complexity by writing f_1, f_2, f_3, and f_4 in the appropriate boxes below.

Lowest asymptotic complexity	f_1
Next highest asymptotic complexity	f_4
Next highest asymptotic complexity	f_2
Highest asymptotic complexity	f_3

12. Consider this method `g`, which operates on an array of integers. The helper method `chooseRandomly` is also defined, as well as the variable `rand`.

```
// A must be non-empty
public static int g(int[] A) {
    int CUT = 3;
    if (A.length < CUT) {
        return A[0];
    }
    else {
        int total = 0;
        for (int i = 0; i < 9; i++) {
            total += g(chooseRandomly(A, A.length / CUT));
        }
        return total;
    }
}

// Random number generator.  The nextInt method runs in constant time.
public static Random rand = new Random();

// Returns an array of k randomly chosen elements from A
// A must be non-empty
public static int[] chooseRandomly(int[] A, int k) {
    int[] B = new int[k];
    for (int i = 0; i < k; i++) {
        B[i] = A[rand.nextInt(A.length)];
    }
    selectionSort(B);
    return B;
}
```

Find a recurrence relation of the form

$$T(n) = a\,T(n/b) + O(n^d)$$

that gives a tight upper bound on the amount of time required to compute $g(A)$ on an n-element array A. Below, give your values for a, b, and d. Also, use the Master Theorem to find a tight $O()$ bound for the recurrence.

	Answer
a	9
b	3
d	2
Bound	$O(n^2 \log n)$

13. Suppose that the first statement of the method `g` in the previous question were modified to store 10 into `CUT`, and that the statement `selectionSort(B)` were removed from the method `chooseRandomly`.

With these two changes in place, repeat the analysis from the previous problem.

	Answer
a	9
b	10
d	1
Bound	$O(n)$

14. You have two algorithms, A and B, for solving a problem involving an array. You know that A is faster than B for arrays longer than d, and that B is faster than A for arrays smaller than d, where d is a positive integer. For each situation below, indicate whether A and B can be blended (using the blending technique discussed in class) to obtain a new algorithm that is faster than both A and B.

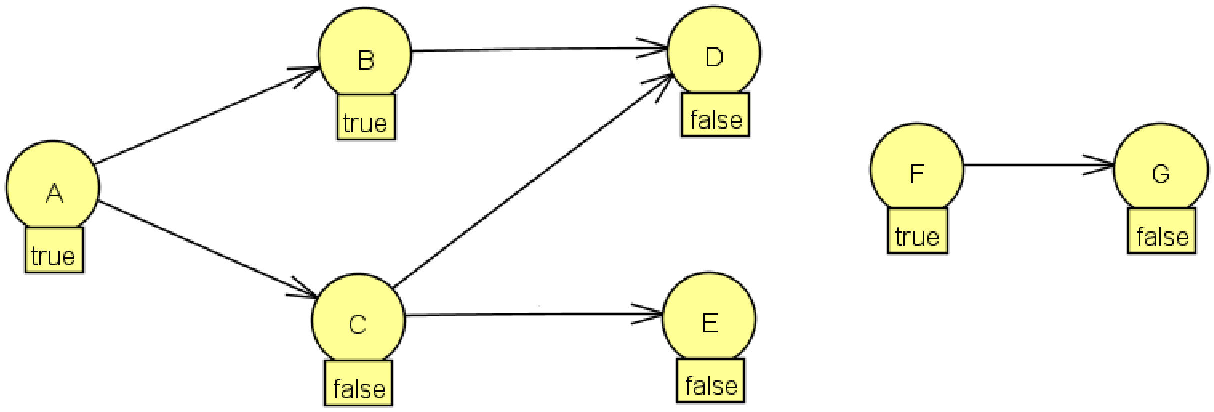
- Situation 1. Both A and B use divide and conquer.
- Situation 2. Neither A nor B uses divide and conquer.
- Situation 3. A uses divide and conquer but B does not.
- Situation 3. B uses divide and conquer but A does not.

Situation	Can be profitably blended (yes or no)
1	yes
2	no
3	yes
4	no

15. [6 points] In this question we will use binary notation. Suppose that the binary numbers 1011 and 1101 are multiplied using the Karatsuba multiplication algorithm. What three pairs of numbers are (recursively) multiplied at the very top level of the recursion tree?

One pair of binary numbers	10 and 11
Another pair of binary numbers	11 and 01
One more pair of binary numbers	101 and 100

16. [6 points] In this question and the next you will be designing an algorithm for directed acyclic graphs, such as the one below:



Let `singleton[u]` be true if vertex u or one of its descendants has exactly one edge leaving it, and false otherwise. The graph above is annotated with the solution for each of its vertices. For example, `singleton[A]` is true and `singleton[C]` is false.

Let `edges[u]` be an array consisting of the vertices reachable from u via exactly one edge. In the graph above, for example, `edges[A] = {B,C}` and `edges[E] = {}`.

We need an algorithm to efficiently compute the value of `singleton[u]` for each vertex u of any dag.

Below, complete the pseudo-code for a method that takes a vertex u of a graph as its parameter and stores a value into `singleton[u]`.

Your method must be implemented so that when it is applied to each vertex of a graph, in an order you will specify in the next question, the entire singleton array will be filled in correctly.

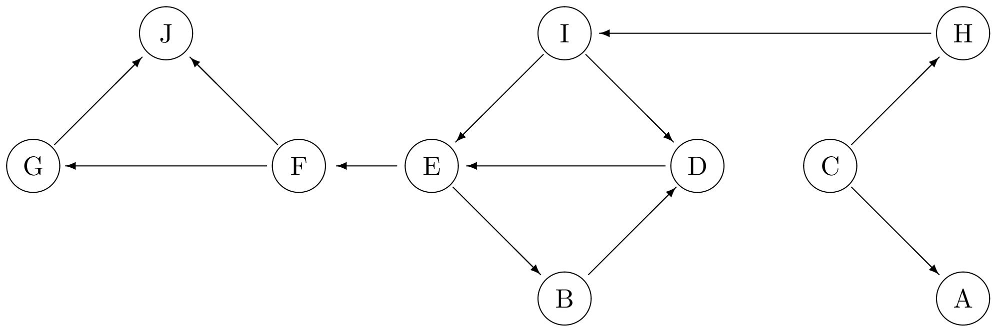
Assume that the `singleton` and `edges` arrays already exist and are accessible to your method. Your method must be $O(n)$, where n is the length of `edges[u]`.

```
void calculate (int u)
  singleton[u] = edges[u].length == 1
  for (int v: edges[u])
    singleton[u] = singleton[u] || singleton[v]
```

17. In the box below, specify the order in which your method from the previous question must be applied to the vertices of a dag.

Answer	reverse topological order

The directed graph X diagrammed below is the subject of the next four questions.



18. Suppose that you do a breadth-first search of X beginning at vertex H.

How many vertices are incorporated into the resulting BFS tree?	8
How many edges are in the longest path of the resulting BFS tree?	4

19. Suppose that you do a depth-first search of X. Suppose also that whenever there is an arbitrary choice of vertices to visit (in either `dfs` or `explore`), you always pick the one that comes first in the alphabet. Give the pre and post numbers requested below. (The lowest pre number should be 1.)

Pre number of B	3
Post number of C	20
Pre number of D	4
Post number of E	12

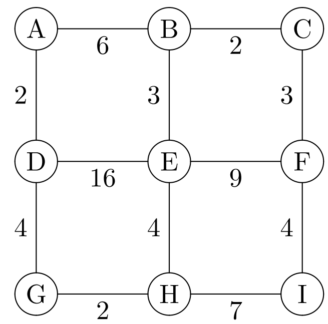
20. This problem concerns the depth-first search described in the previous problem. Give the additional pre and post numbers requested below.

Pre number of G	7
Post number of H	19
Pre number of I	17
Post number of J	9

21. How many strongly connected components does X have?

Answer	8

The next three questions concern this undirected, weighted graph Y.



22. Suppose Dijkstra's algorithm is used on Y to find a shortest paths tree rooted at vertex E. The first vertex to be removed from the priority queue via a `deleteMin` operation will be E. What will be the third and fifth vertices removed? (If there is a tie, choose the vertex that comes earlier in the alphabet.)

Vertex	Answer
Third vertex	H
Fifth vertex	G

23. Continuing with the previous question, what will be the seventh and ninth vertices removed from the priority queue?

Vertex	Answer
Seventh vertex	A
Ninth vertex	I

24. Suppose that a `dist` array for graph `Y` has `dist[E]` $= 0$, and that all the other entries are ∞.

Suppose that the update operation is called on the edges BA, EB, BA, DG, and ED, in that order. Below, give the resulting values of four elements of `dist` by completing the table below.

Entry	Value
`dist[A]`	9
`dist[B]`	3
`dist[D]`	16
`dist[E]`	0