University of Utah
School of Computing

CS 4150 **Final Exam Solution** **April 29, 2011**

The average grade on the final was 66. You can pick up your graded final from the CS office beginning Monday afternoon.

1. [8 points] Use the Master Theorem to derive a big-O bound on the following four recurrences. In each case, $T(1) = O(1)$.

Recursive case	Upper bound
$T(n) = 8T(n/3) + O(n^2)$	$O(n^2)$
$T(n) = 9T(n/3) + O(n \log n)$	$O(n^2)$
$T(n) = T(n/2) + O(1)$	$O(\log n)$
$T(n) = 2T(n/2) + O(1)$	$O(n)$

2. [8 points] Suppose that you want to create an RSA public/private key pair. You begin with two prime numbers, $p = 5$ and $q = 13$. Answer the following questions:

Question	Answer
What is N?	65
What is the smallest possible value for e?	5
Using this value for e, what is d?	$5^{-1} (\text{mod } 48) = 29$
What is the result of encrypting 3?	$3^5 \text{ mod } 65 = 48$

3. [7 points] Write a recurrence relation that expresses the running time $T(n)$ of the method f below, where n is the length of A and A is non-empty.

```
int f (int[] A) {
  if (A.length == 1) {
    return A[0];
  }
  else {
    int[] A1 = new int[A.length/2];
    for (int i = 0; i < A1.length; i++) {
      A1[i] = A[i];
    }
    int[] A2 = new int[A.length - A.length/2];
    for (int i = 0; i < A2.length; i++) {
      A2[i] = A[A1.length+i];
    }
    return Math.min(f(A1), f(A2));
  }
}
```

$T(n) =$	$2T(n/2) + O(n)$
$T(1) =$	$O(1)$

4. [6 points] Indicate whether or not each of the following statements is true or false. Assume that all graphs are directed and unweighted.

1. The meta-graph obtained by running the strongly connected components algorithm will always have a single source and a single sink.

   False. There will always be at least one source and one sink, but there can be more.

2. Following a depth-first search, it is possible for one vertex to have pre/post times of 5/17 and another vertex to have pre/post times of 10/28.

   False. This is impossible. See the table on page 89 of the text.

3. The depth-first search algorithm may fail to visit every vertex.

   False. The DFS algorithm calls explore repeatedly until all vertices are visited.

5. [8 points] Indicate whether or not each of the following statements is true or false. Assume that all graphs are undirected, connected, and weighted, and that all edge weights are positive.

- There is a graph such that when Dijkstra's algorithm and Prim's algorithm are both started at the same vertex, the first edge chosen by Dijkstra's algorithm is different from the first edge chosen by Prim's algorithm.

  True. If the start vertex has two edges with the same smallest weight, Dijkstra's and Prim's can make different choices. Absent ties, however, they will always make the same initial choice. This was a bit tricky, though, so we accepted either answer here.

- There is a graph for which the Bellman-Ford algorithm will perform fewer update operations than Dijkstra's algorithm.

  False. The Bellman-Ford algorithm will update each edge once per iteration, and there will be at least two iterations. Dijkstra's algorithm updates each edge once.

- There is a graph such that the total weight of the edges found by the Bellman-Ford algorithm can be less than the total weight of the edges found by Kruskal's algorithm.

  False. Both algorithms find spanning trees. However, Kruskal's algorithm finds a minimum spanning tree.

- There is a graph such that the total weight of the edges found by Dijkstra's algorithm can be less than the total weight of the edges found by the Bellman-Ford algorithm.

  True. Two different shortest paths trees can have different total weights.

6. [4 points] If it were proven that every algorithm for factoring an integer n must be $\Omega(n)$ in the worst case, would this imply (circle one):

(a) P = NP

(b) P $\neq$ NP

(c) Neither of the above

P $\neq$ NP. A lower bound of $\Omega(n)$ would imply that factoring requires exponential time, since $n = 2^{\log n}$ is exponential in the length of n.

7. [4 points] If an $O(n^{100})$ algorithm for solving the subset sum problem were discovered, where n is the total number of bits required to represent the set, would this imply (circle one):

(a) P = NP

(b) P $\neq$ NP

(c) Neither of the above

P = NP. This discovery would be of a polynomial-time algorithm for an NP-complete problem.

8. [4 points] If it were proven that every algorithm for solving the Hamiltonian (Rudrata) cycle problem on a connected graph must be $\Omega(2^E)$ in the worst case, where E is the number of edges in the graph, would this imply (circle one):

(a) P = NP

(b) P $\neq$ NP

(c) Neither of the above

P $\neq$ NP. We would have proven that not all the problems from NP can be solved in polynomial time.

9. [4 points] Suppose that A and B are search problems, and A is known to be NP-complete. You want to prove that B is also NP-complete. Which of the following should you do (circle one)?

(a) Show that there is a polynomial time reduction from A to B.

(b) Show that there is a polynomial time reduction from B to A.

(c) Find an exponential time algorithm for A.

(d) Find an exponential time algorithm for B.

Reduce from A to B.

10. [6 points] Solve for x in each of the modular equations below. In each case, x must be positive (greater than zero) and less than the modulus.

Equation	Solution
$5 \cdot x \equiv 2 \pmod{7}$	6
$x \cdot x \equiv 0 \pmod{8}$	4
$7^{-1} \equiv x \pmod{11}$	8

11. [4 points] Here are four of the steps in a secure electronic commerce transaction:

(a) The browser encrypts something with an AES key

(b) The browser encrypts something with an RSA public key

(c) The browser requests a digitally signed certificate

(d) The browser decrypts something with an RSA public key

Put the four steps in order, beginning with the one that happens first and ending with the one that happens last.

(c) We request a signed certificate from the server.

(d) We verify (decrypt) the certificate with the certifying authority's public key.

(b) We generate an AES key, encrypt it with the server's public key that was contained in the certificate, and send it to the server.

(a) We communicate by encrypting with the AES key.

12. [6 points] Suppose that you need an algorithm that can find the k smallest elements in an unordered array of n integers. Give a tight upper bound (in terms of k and n) on the *expected running time* for both of the following algorithms.

Algorithm	Upper Bound
Sort the array using quicksort. Copy out the first k elements.	$O(n \log n)$
Use quickselect (the randomized selection algorithm that we studied) to find the k^{th} smallest element. Use this element as a pivot and partition (using the partitioning algorithm from quicksort) the array. Copy out the first k elements.	$O(n)$

Keep in mind that we were looking for a bound on the expected (average) running time.

13. [8 points] Each row of the table below gives bounds on $f(n)$ and $g(n)$. At the end of each row, indicate whether the bounds on that row imply that g is $O(f)$), f is $O(g)$, or neither.

Bound on f	Bound on g	Relationship
f is $O(n^2)$	g is $O(n^3)$	neither
f is $O(n^2)$	g is $\Omega(n^3)$	f is $O(g)$
f is $\Theta(n^2)$	g is $O(n^3)$	neither
f is $\Theta(n^2)$	g is $\Theta(n)$	g is $O(g)$

Keep in mind that big-O bounds are not necessarily tight.

14. [4 points] Suppose that you have a rope that is m meters long and that contains n knots. You want to cut the rope into the smallest possible number of pieces such that no piece has two knots that are more than one meter apart. (All cuts must be made *between* knots.) Is this problem best solved via dynamic programming, a greedy algorithm, or divide and conquer?

Answer	Greedy

Simply cut between the knots that are more than one meter apart.

15. [4 points] Suppose that a weighted, undirected, connected graph contains V vertices and $\Theta(V \log V)$ edges. Using O notation, give a tight upper bound on the time complexity of Dijkstra's algorithm on such a graph. Assume that the graph is represented with an adjacency list and that the priority queue is represented with a binary heap.

Answer	$O(V \log V \log V)$

16. [4 points] You are the commander of an army. At 8:00 a.m. you intercept an encrypted message intended for the enemy commander. The message is encrypted via RSA using the enemy commander's public key. You know his public key, but unfortunately you do not know his secret key. The key pair was constructed using 8192-bit prime numbers.

Your spies have determined that the original message was either "Attack from the east at noon" or "Attack from the west at noon". Is it reasonable to expect that your encryption specialist will be able to figure out what the original message was in the four hours remaining before the attack begins?

Answer	yes

Simply encrypt both of the possible messages and see which one yields the enrypted message that you intercepted.

17. [11 points] Consider the problem of determining whether a string S consists of a sequence of English words. For example, "blackandblue" does, but "blackandblu" does not.

Here is a greedy algorithm for this problem:

- If S is empty, report true.
- If S does not begin with an English word, report false.
- Otherwise, remove a word with which S begins and return the result of running the algorithm on the reduced version of S.

(a) Unfortunately, this algorithm does not work in all cases. Give a string on which the algorithm will fail.

Answer	an

Having failed to find a greedy algorithm, let's look for a dynamic programming algorithm. Let's assume:

- That S contains n characters, S[1] through S[n].
- That there is a function dict(S, i, j) that returns true if and only if the substring S[i..j] is an English word.

Let's define W(i) to be true if the characters S[1] through S[i] consist of a sequence of English words. Here's the skeleton of a recursive definition of W(i).

```
W(0)  =   _____


W(i)  =     or       ( W(_____) and dict(S, _____, _____) )
        0 <= j < i
```

The idea for the recursive case is to try out all possible places for the final word break in S[1..i].

(b) There are four blanks in the solution. Going top to bottom and left to right, how should the blanks be filled in?

Blank	Answer
1	true
2	j
3	j+1
4	i