

University of Nevada, Las Vegas Computer Science 477/677 Spring 2012

Examination, February 23, 2012

The entire examination is 270 points.

Name: _____

No books, notes, scratch paper, or calculators. Use pen or pencil, any color. Use the rest of this page and the backs of the pages for scratch paper. If you need more scratch paper, it will be provided.

1. True or False. [5 points each]

- (a) _____ Computers are so fast today that complexity theory is only of theoretical, but not practical, interest.
- (b) _____ If any problem can be precisely formulated in a mathematical way, there is an algorithm that solves it.
- (c) _____ If S is a set of distinct items, we say that an $x \in S$ has *rank* k if there are exactly k members of S which are less than or equal to x . If, while implementing quicksort to sort a set of n distinct items, we always pick the pivot (cut) item to be an item whose rank is at least 10% of the size of the subset we are currently sorting, and never more than 90% of the size of that subset, the time complexity of our implementation will be $\Theta(n \log n)$.
- (d) _____ Any comparison-based sorting algorithm must use at least $\lg(n!)$ comparison to sort n items, in the worst case.
- (e) _____ There exists an $O(n)$ -time algorithm to find the median of a set of n numbers.

2. Fill in the blanks. [5 points each blank.]

- (a) _____ works by solving a sequence of subproblems in bottom-up order.
- (b) A _____ algorithm always makes that choice which results in the greatest immediate gain.
- (c) On a directed weighted graph G of n nodes and m edges, where no edge has negative weight, we can solve the single source shortest path problem on G in _____ time. (Give an asymptotic answer).

3. Solve the recurrences. Give asymptotic answers in terms of n , using either O , Ω , or Θ , whichever is most appropriate. [10 points each.]

(a) $F(n) = 4f(n/2) + n^2$.

(b) $G(n) \geq G(n-1) + \lg n$

(c) $H(n) \leq 2H(\sqrt{n}) + 4$.

(d) $K(n) = K(n - 2\sqrt{n} + 1) + n$.

(Warning: do not attempt this one until you are finished with the rest of the test. I do not expect more than two students to get it.)

4. [15 points] Consider the following procedure:

```
void george(int n)
{
    & int m = n;
    & while (m > 1)
    & {
    &     for (int i = 1; i < m; i++)
    &         cout << "I cannot tell a lie.  I chopped down the cherry tree." << endl;
    &     m = m/2;
    & }
}
```

Consider the question of how many lines of output execution of `george(n)` would produce. Write down an appropriate recurrence for this question, and give an asymptotic solution in terms of n , using either O , Ω , or Θ , whichever is most appropriate.

5. [20 points] Sort the functions given below in order of increasing asymptotic complexity. If two or more functions have the same asymptotic complexity, indicate that.

n , $5n + \sqrt{n}$, n^{100} , $\log_2 n$, $\log_3 n$, $n^{0.001}$, 1.001^n , $\log_2(n!)$, $n \log_2 n$.

6. [30 points] Consider the SPLIT procedure used in Quicksort, also known as PARTITION, or (in our textbook) PIVOT. Write pseudo-code for that procedure (not any of the other parts of quicksort) and write a loop invariant for your code.

7. [30 points] Find the longest common subsequence of the two sequences:

abracadabra

crabnebula

8. [30 points] Use Huffman's algorithm to find a prefix code for the alphabet **A B C D E F** where the frequencies are given in the following table:

A	10
B	5
C	6
D	12
E	20
F	2

9. [30 points] You are given $n \times n$ 2-dimensional array, $W[i, j]$ representing the weights of the edges of a weighted directed graph G . You may assume that If there is no edge from i to j in G , then $W[i, j] = \infty$. Write code (or pseudo-code) for a program that solves the all-pairs shortest path problem for G , using the Floyd-Warshall algorithm.
- Your code should contain a portion which prints an error message if G has a negative cycle.

10. [30 points]

Assume you are given an array containing the following letters: **R E Y A M T K C W Z G**.

Using the version of heapsort we learned in class, indicate the contents of the array each time there is any change. Use the next page if necessary.

[illegible]

