Computer Science 302 001 Spring 2005 Final Examination, May 12, 2005

Name:____

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

The entire examination is NNN points.

1. True or False. [5 points each]

   (a) ________ In the decision tree model of computation, the time complexity of any algorithm to sort n items is $\Omega(n \log n)$.

   (b) ________ The height of a binary tree with n nodes is $\Omega(\log n)$.

2. [10 points] What implementation of the ADT *search structure* would you use if the expected number of items in the structure is 1?

3. [10 points] What implementation of the ADT *search structure* would you use if n items are to be inserted at once at the beginning of the program, there will be no further inserts, and *find* will be executed n^2 times during the running of the program? (There is more than one correct answer to this problem, as well as several inferior answers.)

4. [10 points] Suppose an $5 \times 6 \times 8$ array `A` is stored in row-major order, with base address 1000, and one word is required for each entry. What is the address of `A[3,1,6]`? Assume that array indices start at 0, as in C++.

5. [20 points] Write pseudocode for the array implementation of the ADT "stack of integer." Your code should include procedures that implement pop, push, and empty.

6. [20 points] Explain how you would insert and delete from a queue, given that you are using singly linked nodes in a circular linked list implementation.

7. [20 points] Using min-plus matrix multiplication, compute AB, where A and B are the two matrices given below:

8. [20 points] By hand, walk through the algorithm Graham Scan to find the convex hull of the set of points in the plane shown below. Mark up the figure to show what you are doing, as needed.

9. Describe each of the following types of search. (Be sure to say what the structure is that is being searched in each case.)

   (a) Linear search.

   (b) Binary search.

   (c) Depth first search.

   (d) Breadth first search.

10. Give four ways to implement the ADT "search structure." (For example, "binary search tree" is one way, so don't use that one.) Just give the names of the implementations, not any details.

11. Give two radically different data structures that could be used to represent a directed graph. Which one is usually better if the graph is sparse?

12. You can sort a set of items using a binary search tree, as follows: Start with an empty binary search

tree, insert the items one at a time, then visit the nodes of the tree in inorder, writing out the items. This algorithm is essentially the same as which one of these well-known sorting algorithms? (Choose one answer.)

(a) Quicksort

(b) Heapsort

(c) Mergesort

(d) Bubblesort

13. [10 points each]

For each of the following code fragments, express the asymptotic time complexity by choosing the best of the following answers: $O(n)$, $O(n^2)$, $O(n \log n)$, $O(\log n)$, $\Theta(n)$, $\Theta(n^2)$, $\Theta(n \log n)$, $\Theta(\log n)$,

(a)
```
for (int i = 0; i < n; i++)
  cout << "Hi there.";
```

(b)

(c)
```
for (int i = 0; i < n; i = 2*i+1)
  cout << "Hi there.";
```

```
for (int i = 0; i < 0; i++)
  for (int j = i; j > 0; j = j/2);
  cout << "Hi there.";
```

(d)
```
for (int i = 0; i < n; i++)
  for (int j = i; j > i/2; j = j/2);
    cout << "Hi there.";
```

14. [20 points]

Summarize the program you wrote for Assignment 6. Your summary should (approximately) fill two pages. In this summary, you should explain what data structures you used for each part of the program, how you used them, and how they were implemented.

15. [30 points]

(a) Describe the meaning of the word **collision** as used in discussions of hashing.

(b) How are collisions handled in **closed** hashing?

(c) How are collisions handled in **open** hashing?

(d) What is **perfect** hashing, and under what conditions can it be used?

16. [25 points] Explain how insertion works in a B-tree. (You may assume that it is a B*-tree or a B+ tree if you wish.) Hint: the phrase "node splitting" or the equivalent must be in your explanation.

17. [20 points] Write pseudocode for the algorithm *binary search*.

18. [20 points] Write C++ code for the *find* portion of union-find as you used in Assignment 6. Be sure to copy the code that you wrote for that assignment as precisely as you can, using the same variable names. (I hope you used path compression as you were supposed to.)

    Do not include any other part of the program. If you write more than 10 lines, you've written far too much.

19. [30 points]

    Describe the algorithm *polyphase mergesort*. You should be able to fit the answer onto just this page, but you can use the back of the page if you need to.

20. Give the matrix representing the weighted directed graph G shown in Figure 1. Then use matrix multiplication to find the matrix representing the solution to the all-pairs shortest path problem in G.

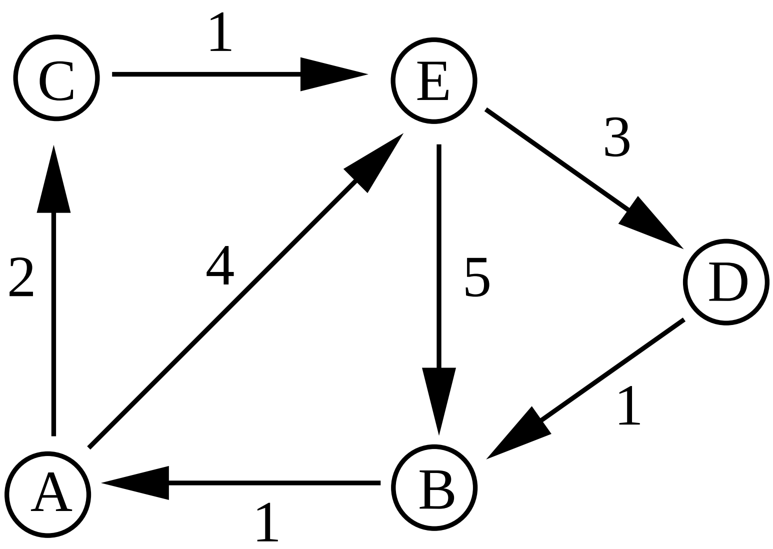

Figure 1: Weighted Directed Graph for Problem 20

21. Find the solution to the single-source minpath problem, where the source is A, for the weighted directed graph shown in Figure 2, using Dijkstra's algorithm.

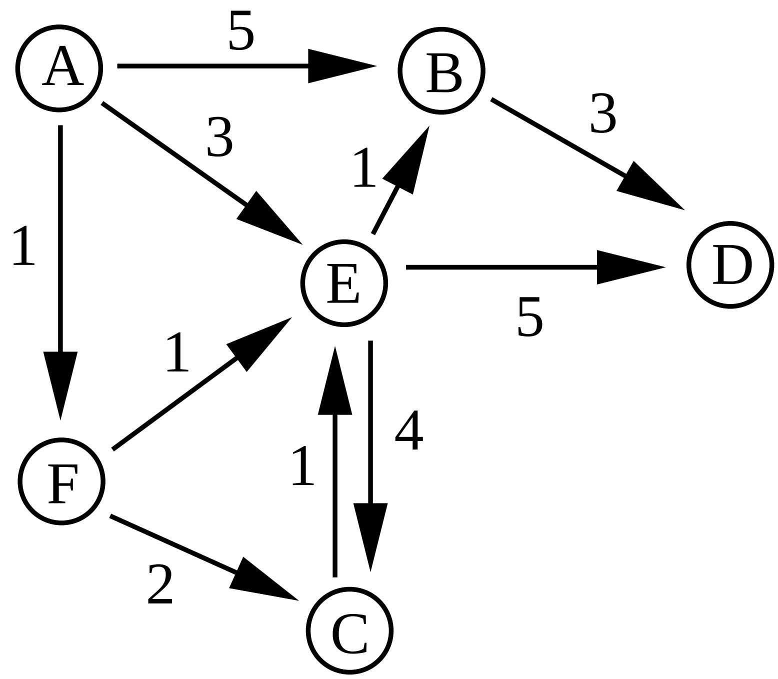

Figure 2: Weighted Directed Graph for Problem 21