Computer Science 302 Fall 2018 (Practice for) Third Examination, November 14, 2018

Name:_

The entire practice examination is 400 points.

- 1. True or False. [5 points each]
 - (a) _____ A good programmer should never use linear search.
 - (b) _____ The height of a binary tree with n nodes is $O(\log n)$.
 - (c) $\dots 3^n = O(2^n).$
 - (d) _____ The height of a **balanced** binary search tree with n items is $O(\log n)$.
 - (e) $\ldots \log \log n = O(\log n).$
 - (f) _____ A stack is an example of a search structure.
 - (g) _____ A treap is an example of a search structure.
 - (h) _____ An item can always be inserted into a min-heap of size n in $O(\log n)$ time.
 - (i) _____ A treap is an example of a priority queue.

2. Fill in the blanks.

- (a) For each of the following blanks, the correct answer is stack, queue, array, heap, or search structure.(5 points each blank.)
 - i. [5 points] *push* is an operator of _____.
 - ii. [5 points] find is an operator of _____.
 - iii. [5 points] *fetch* is an operator of _____.
 - iv. [5 points] deletemin is an operator of _____.
 - v. [5 points] You would use a _____ to do breadth first search of a graph.
 - vi. [5 points] You would use a _____ to hold the records of the customers of a business.
 - vii. [5 points] You would use an _____ to keep track of the number of times each student in a class asks a question.
 - viii. [5 points] You would use a _____ to keep track of patients in the waiting area of the emergency room of a hospital.
 - ix. [5 points] You would use a ______ to match left with right parentheses in an algebraic expression.

- x. [5 points] You would use a _____ to store your unpaid bills, if every time you get money, you pay as many bills as possible, in order of urgency.
- xi. [5 points] In the decoding phase of the Huffman assignment, you needed to write as many words as possible on each line without overflowing the line. You used (or should have used) a _________ algorithm for this task. (The choices are greedy, divide and conquer, dynamic programming.)
- 3. [20 points] Build a treap by entering items in the order listed below. Use min-heap order for the random keys. The first column shows the item, the second column the random key. Show the treap after each insertion, and show the rotation steps.

| Dan | 3 |
|-----|---|
| Bob | 4 |
| Jil | 5 |
| Ann | 2 |

- 4. State the asymptotic complexity of each of the following code fragments in terms of n, where n is a positive integer. Use Θ notation.
 - (a) [10 points]

```
for(int i=0; i<n; i++)
for(int j=n+1; j>i; j=j-2)
    cout << "Hello world" << endl;</pre>
```

(b) [10 points]

```
for(int i=1; i<n; i++)
for(int j=0; j*j<i; j++)
    cout << "Hello world" << endl;</pre>
```

(c) [10 points]

```
void george(int n)
{
  for(int i = 0; i < n; i++)
    cout << "Hello George" << endl;
    if(n > 1)
      {
      george(n/2);
      george(n/2);
    }
}
```

5. [20 points] Consider the following binary tree T.



List the nodes of T in preorder, inorder, postorder, and level order.

6. [30 points] Consider the heap of 12 elements, implemented as an array as illustrated below. Illustrate the implementation after insertion of the letter **E**. (You are not required to show steps, but it doesn't hurt.)

| B | Μ | K | Q | N | S | L | Z | W | Р | R | U | X | B | V | W | K | A |
|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | | | | | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | | | | | | |

7. [30 points] Consider the treap illustrated below, where the heap key is a randomly chosen integer in the range 0...99. A new item, "Fay," is inserted, and the heap key "12" is randomly chosen. Show the treap after that insertion, and show the intermediate steps.



8. (a) [10 points] Write a topological ordering of the nodes of the weighted directed graph shown in the figure below.



(b) [20 points] Use dynamic programming to solve the single source shortest problem on that graph.

9. [20 points] You are building a cuckoo hash table for the following data set consisting of 8 items. The indices of your has table are $\{0, 1, \ldots, 9\}$. The two hash values for each item are listed in the first and second columns of the array below.

Walk through the steps of inserting the items, in the order given in the array.

| Ann | 1 | 4 |
|-----|---|---|
| Dan | 8 | 7 |
| Kim | 2 | 8 |
| Sam | 5 | 2 |
| Zoe | 3 | 9 |
| Ted | 5 | 0 |
| Kat | 8 | 2 |
| Max | 3 | 4 |

10. [20 points] Use the Union/Find data structure to find the components of the graph whose vertices are the integers in the range 0...4 and which has three edges, namely $\{0,1\}$, $\{2,3\}$ and $\{1,4\}$.

- 11. [20 points] A is a two-dimensional rectangular array with row indices 0..8 and column indices 0..6 We write A[i][j] for the item in row i and column j.
 - (a) If A is stored in main memory in row-major order with base address 1000, and if each entry takes one word in main memory, find the address in main memory of A[7][3].
 - (b) If A is stored in main memory in column-major order with base address 2000, and if each entry takes two words in main memory, find the address in main memory of A[7][3].

- 12. [20 points] Find an optimal prefix code for the alphabet $\{A, B, C, D, E, F, G, H\}$, if the frequencies of the symbols are as given in the following table:
 - A = 35
 - B = 7
 - C = 32
 - D = 5
 - *E* 16
 - F = 4
 - G = 11
 - H = 5

13. [20 points] Indicate the minimum spanning tree of the following weighted graph.



14. [20 points] In Union/Find, the following code implements Find, but does not compress paths. Rewrite the code using path compression.

```
int parent[N]; // i is a leader if parent[i] = i
int find(int i)
{
    if (parent[i] == i)
      return i;
    else
      return find(parent[i]);
}
```

15. [20 points] The following code computes a function F(n) for any positive integer n, using recursion. However, it takes an unnecessarily long time to run. Write a dynamic programming algorithm which computes F[n] for any positive integer n. What is the asymptotic time complexity of your code, in terms of n?

```
int F(int n)
{
    if(n < 1) return 1;
    else return 1 + F(n-1) + F(n/2);
}</pre>
```

۷