Computer Science 302 Spring 2018 (Practice for) Final Examination, May 9, 2018

Name:____

The entire practice examination is 880 points.

1. True or False. [5 points each]

   (a) ________ The time to heapsort an array of n items is $O(n \log n)$.

   (b) ________ Open hashing uses open addressing.

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

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

   (e) ________ A binary search tree is commonly used to represent unfulfilled obligations.

   (f) ________ An acyclic directed graph is always a tree.

   (g) ________ Quicksort takes $O(n \log n)$ average time to sort an array of n items if the pivots are picked at random.

   (h) ________ Given the choice between two algorithms, one of which takes $O(n)$ time and the other of which takes $O(n^2)$ time, is it *always* best to choose the one which takes $O(n)$ time?

   (i) ________ Computers are so fast nowadays that, as a practical matter, we should not worry about the time complexity of a program.

2. [10 points] What is the relationship between the number of vertices and the number of edges of a planar graph?

3. [10 points] The vertices of a directed graph G are in ______________________________ order if x comes before y for every directed edge (x, y) of the graph. If there is such an order, G must be ______________________________.

4. [10 points] Here is a recursive function for the n^{th} Fibonacci number for any positive integer n.

```
int fibonacci(int n)
 // input condition: n > 0
 {
  if(n <= 2)
   return 1;
  else
   return fibonacci(n-2)+fibonacci(n-1);
 }
```

   Is this function correct? Would it be a good idea to use it? Why not?

5. [10 points] A connected acyclic graph of n vertices has ______________________________edges.

6. [10 points] What search structure should you use if the average number of items that will be in the structure at any given time is two?

7. [5 points] The items in a ______________________________typically represent unfulfilled obligations.

8. [10 points] The two operators of the ADT *array* are ______________and ______________.

9. [10 points each] For each of the following code fragments, assume that a positive integer **n** is given. State the asymptotic time complexity in terms of **n**, using Θ notation if possible.

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

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

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

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

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

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

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

10. [30 points]

(a) In hashing, what do we mean by a "collision"?

(b) How are collisions handled in closed hashing?

(c) How are collisions handled in open hashing?

11. [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.)

12. [20 points] Walk through the steps of the stack algorithm used to evaluate the following postfix expression, showing the stack at each step: (Hint: there will be approximately 9 illustrations of the stack.)

```
5 6 + 3 * 2 3 * -
```

13. [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

14. [30 points] The *Partition* step of Quicksort has a loop invariant. Give that loop invariant, and illustrate its meaning by drawing a figure, or figures.

15. [30 points] 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.

16. [20 points] The following is an array implementation of a stack of floats. Finish the function which pushes a new float onto the stack.

```
struct stack
 {
  float item[100];
  int top;
 }

void push(stack*mystack,float newitem)
 {
  assert(top <   ); // what constant goes here?


 }
```

17. [10 points] We have an assert statement in the push function in problem 16 above because C++ does not have bounds checking. What should we write as the argument of that assertion?

18. [20 points] Given the following implementation of a binary tree, complete the recursive function which writes the items of a tree in postorder.

```
struct treenode
 {
  int item;
  treenode*left;
  treenode*right;
 };

void postordervisit(treenode*root)
 // uses recursion
 {
  if(root)
   {



   }
 }
```

19. [20 points] Given the following linked list implementation of a stack, complete the function which implements pop.

```
struct stacknode
 {
  float item;
  stacknode*link;
 };

float pop(stacknode*&mystack)
 {
  assert(mystack);
 }
```

20. [10 points] What is the purpose of the assertion in the function in problem 19 above?

21. [10 points] Suppose you are writing a dynamic programming algorithm to find the minimum weight path between a given source vertex S and a given target vertex T in a weighted directed acyclic graph G.

   (a) Describe the subproblems.

   (b) In what order would you work the subproblems?

22. True or False. [5 points each]

    (a) ________ Quicksort takes $O(n \log n)$ expected time to sort an array of n items, provided randomization is used to pick the pivot items.

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

23. [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.)

24. [30 points] Describe each of the following types of search. (Be sure to say what the structure is that is being searched in each case.)

    (a) Breadth first search.

    (b) Depth first search.

25. [40 points]

    (a) What is the ADT "search structure"? Give three examples.

    (b) What is the ADT "priority queue"? Give three examples.

26. [20 points] Explain "cuckoo hashing."

27. [10 points] Binary tree sort is actually another way to implement which one of the following three standard sorting algorithms?

    (a) Quicksort

    (b) Heapsort

    (c) Mergesort

28. [10 points] Heapsort is actually a fast way to implement which one of the following three quadratic time sorting algorithms?

    (a) Bubblesort

    (b) Insertion sort

    (c) Selection sort

29. [20 points] Write C++ code for the **find** portion of union-find. Be sure to use path compression. Do not include any other part of the program. If you write more than 10 lines, you've written far too much.

30. [10 points each] For each of the following code fragments, express the asymptotic time complexity by choosing the best of the following answers: $\Theta(n)$, $\Theta(n^2)$, $\Theta(n \log n)$, $\Theta(\log n)$, $\Theta(\log \log n)$.

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

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

(c)
```
for (int i = 0; i < n; 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.";
```

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

31. [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?

32. [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.)

33. [20 points] Explain how you would implement a sparse array using a search structure. Do **not** give any details whatsoever about the search structure itself, since that's not the point of this question.

34. [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. Draw pictures.

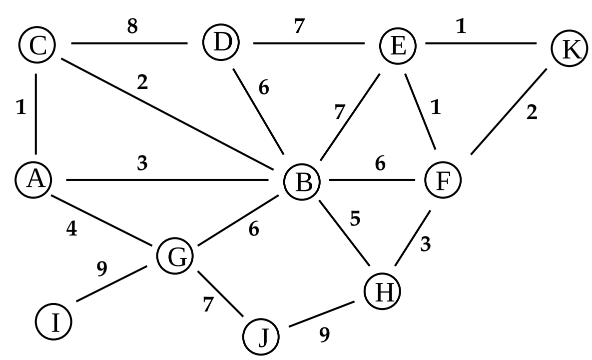
35. [30 points] Use polyphase mergesort to sort the following list: FUNWITHPOLYPHASE Show all steps.

36. [30 points] `A` is a $4 \times 8 \times 5$ array and is stored in RAM in column major order, with base address 2048. Each entry of `A` is stored in two address locations of the RAM. Calculate the base address of `A[2][5][3]` in RAM. As in C++, assume that the first value of each index of `A` is 0.

37. [30 points] Sort the following array using Heapsort, showing the array after each step. For your convenience, I have included a figure to make it easier for you to write those arrays. The number of rows in the figure below may or may not be equal to the number of steps; you might not use all the rows, or you might have to add more rows.

A	L	G	O	R	I	T	H	M

38. [20 points] Find a minimum spanning tree of the weighted graph shown below. You need not show work, just indicate by darkening edges.



39. [40 points] Write a complete C++ program that reads a file of integers, two integers on each line, and prints the sum of those integers. You may assume that the program is executed by typing

`./a.out < infile > outfile`.

40. [20 points] The loop invariant of the loop in the following function is `x*y+z == n*m`. What is the purpose of this function? How does the loop invariant allow us to prove correctness?

```
int product(int n, int m)
 // input condition: m >= 0
 {
  int x = n;
  int y = m;
  int z = 0;
  // Loop invariant: x*y + z == n*m holds here
  while (y > 0)
   {
    // Loop invariant holds here
    x = 2*x;
    // Loop invariant does not hold here
    if(y%2) z = z+x;
    // Loop invariant does not hold here
    y = y/2;
    // Loop invariant holds here
   }
  // Loop invariant holds here, which allows us to prove correctness
  return z;
 }
```

41. [20 points] What is the purpose of the following function? What is the loop invariant?

```
float power(float n, int m)
 // input condition: m >= 0
 {
  float x = n;
  int y = m;
  float z = 1;
  // Loop invariant holds here
  while (y > 0)
   {
    // Loop invariant holds here
    x = x*x;
    if(y%2) z = z*x;
    y = y/2;
    // Loop invariant hold here
   }
  // Loop invariant holds here, which allows us to prove correctness
  return z;
 }
```