Multiple choice – 3 points each.

1. Consider the following function:
   
   ```c
   int koodle ( int n )
   {
   int j, k; bool x, y;
   if ( n % 2 )
       for ( j = 0; j < n; j++ )
           x = rand() % 2;
   else
       for( k = 0; k < n ; k++ )
           y = rand() % 2;
   }
   // koodle
   ```

   The complexity of this function is:
   a. O(n)
   b. O(n log n)
   c. O(n^2)
   d. O(2^n)
   e. None of the above

2. Consider the following recursive function:
   
   ```c
   void qq(int n)
   {
   int i; int x;
   if ( n <= 0 ) return;
   x++; //do something
   qq(n-1);
   qq(n-1);
   }
   // qq
   ```

   The complexity of this function is:
   a. O(n)
   b. O(n log n)
   c. O(n^2)
   d. O(2^n)
   e. None of the above

3. Which is true of binomial heaps as compared to binary heaps?
   a. Findmin is faster in binomial heaps
   b. Findmin in faster in binary heaps
   c. DeleteMin is faster in binomial heaps
   d. Merging is faster in binary heaps
   e. All of the above
4. The best case, in terms of the asymptotic running time, for QuickSort happens when:
   a. The array is already sorted
   b. The array is sorted in reverse
   c. All elements of the array are equal
   d. The number of elements to the right of the pivot is approximately equivalent to the
      number of elements to the left of the pivot
   e. All elements of the array are negative

5. Recall all sorting algorithms you have studied. Suppose that you have an array of 128 elements. Which number is closest to the minimum number of binary element comparisons that would be necessary to sort the elements?
   a. 10
   b. 4075
   c. 389
   d. 891
   e. 7887

6. Consider the following graph:

   ![Graph Image]

   What is the cost of the minimum spanning tree for this graph?
   a. 1
   b. 19
   c. 20
   d. 21
   e. None of the above

7. You write a hash table implementation that uses separate chaining. Which technique is most efficient for handling collisions?
   a. Linear probing
   b. Quadratic probing
   c. Double hashing
   d. None of the above

8. Every node in a (min) binary heap
   a. Has two children
   b. Is no larger than its children
   c. Is no smaller than its children
   d. Has a smaller left child than right child
   e. Two or more of the above
9. Which tree could **NOT** result after deleting 10 from following simple binary search tree:

\[ \text{Diagram of the binary search tree with 10 as the root node.} \]

- a. 
- b. 
- c. 
- d. They could all result from the deletion.
- e. None of them could result from the deletion.
10. For the partial solution to the network flow problem (where edges are marked with current flow/capacity), which of the following is NOT an augmenting path

- a. ABDEG
- b. ABCFG
- c. ABEG
- d. ABDG
- e. None of the above

11. $n$ elements are inserted one by one into an initially empty binary heap (stored as an array). The total running time for all insertions is:
   - a. $O(n)$ worst case and $O(n)$ average case
   - b. $O(n)$ worst case and $O(\log n)$ average case
   - c. $O(n \log n)$ worst case and $O(n)$ average case
   - d. $O(n \log n)$ worst case and $O(n \log n)$ average case

12. What item is the deepest after the following sequence of insertions into an empty splay tree: 1, 11, 3, 10, 8?
   - a. 11
   - b. 3
   - c. 10
   - d. 8
   - e. 1
   - f. Two of the above

13. Which of the statements a to d about splay trees is false?
   - a. A single access operation could examine every node in the tree
   - b. Any $n$ consecutive operations from an initially empty splay tree must take at most $O(n \log n)$ time
   - c. Inserting the items 1, 2, ..., $n$ into an initially empty splay tree takes $O(n)$ total time
   - d. The most recently accessed item is at the root
   - e. None of a to d is false

14. Consider the following recursive function:
    ```c
    void q(int n)
    {     if ( n <= 0 ) return;
         int x=3;  //do something
    ```
The complexity of this function is: 
   a. O(\log n) 
   b. O(n) 
   c. O(n \log n) 
   d. O(n^2) 
   e. O(2^n)

15. A friend explains he has written a greedy algorithm. This implies: 
   a. The algorithm is efficient, but not optimal 
   b. The algorithm is optimal, but not efficient 
   c. The algorithm runs in polynomial time 
   d. An exhaustive search will be performed 
   e. None of the above 

16. Finding a node (in an AVL tree) has what complexity? 
   a. O(n \log n) 
   b. O(n) 
   c. O(\log n) 
   d. O(1) 
   e. none of the above 

17. For the spanning tree below, which edge would NOT be part of a minimal spanning tree. 
   a. GF 
   b. DF 
   c. BC 
   d. AB 
   e. DE 

18. Which is true of a mergesort and a quicksort? 
   a. both have the same average case complexity 
   b. both are greedy 
   c. both take O(n) to combine the pieces 
   d. all of the above
19. Suppose that we are performing double hashing. Our table size is m = 300. Recall that the first hash function gives the initial location to be probed and the second hash function gives the size of the steps we should jump as we follow the probe sequence. Which of the following values for the second hash function will guarantee that, when doing an insertion, we will find an empty table location unless the entire table is full?
   A. 21  B. 34  C. 49  D. 65  E. 98

20. Which of the functions below grows the fastest for large values of n?
   A. \( n^{10} \)  B. \( 3^n \)  C. \( 2^n \)  D. \( 2^{2n^2} \)  E. \( 4^n \)

21. Suppose that we build a splay tree by starting with an empty tree and inserting 21, then 37, then 41, and finally 28. Which choice lists the keys of the resulting tree in preorder?
   A. 41, 37, 21, 28
   B. 21, 28, 37, 41
   C. 28, 21, 37, 41
   D. 28, 21, 41, 37

22. Which of the following is not one of the standard heap operations?
   A. Insert an item with a given value
   B. Delete the item with the minimum value
   C. Search for an item with a given value
   D. All of the above

23. For which of the following min heap implementations did we require that the value in a node be no smaller than the value in its parent?
   A. d-heaps
   B. Binomial heaps
   C. binary heaps
   D. All of the above

24. When discussing binomial heaps we talked about a type of tree called a binomial tree. Suppose that the root r of a binomial tree has 6 children. How many grandchildren does r have? (Note: a grandchild of r means a child of a child of r)
   A. 4  B. 8  C. 10  D. 15  E. 20

25. Consider the “order” of complexity classes from “fastest” to “slowest” in the intuitive sense. Which of the following seems most appropriate?
   (a) \( O(n) < O(1) < O(\log n) < O(n^2) \)
   (b) \( O(1) < O(n) < O(\log n) < O(n^2) \)
   (c) \( O(1) < O(\log n) < O(n \log n) < O(n^2) \)
   (d) \( O(1) < O(\log n) < O(n^2) < O(n \log n) \)
   (e) None of the above.

26. You need a DIGRAPH (directed graph) implementation for large, sparse graphs. Your application checks for edges between nodes most frequently; it also needs to find all nodes
connected to a given node a lot; but it rarely inserts or removes anything. Which of the
following data structures seems most appropriate?
(a) Linked lists used as adjacency lists.
(b) adjacency matrix
(c) hash-tables
(d) All of the above work equally well.
(e) None of the above.

27. Your teacher uses the phrase, “When the doorbell rings, I can clean up things on the way
to the front door, but I am not allowed to clean my windows.” The analogy is
(a) data structures should have a destructor which cleans up any dynamically allocated space
(b) when someone demands service, the data structure has no choice but to provide it
(c) doing extra work along the way does not change complexity. It just changes the constant
multiplier.
(d) linear complexity is achieved by avoiding loops

28. Your instructor says you must have faith your recursive routine works. The reason for
this is
(a) positive thinking is required or you will never begin
(b) you cannot mentally trace a recursive call as the function is not yet written. Since tracing
is not an option, faith is required.
(c) it is impossible to trace the code by hand, so you just have to guess
(d) usually the code is better than you think. Try it before you give up.

29. Your instructor says, “When you ask your little brother to get you a drink of water, it is
rude not to drink it.” The analogy is
(a) When a recursive routine returns a value, if you don’t use it, the code is probably wrong.
(b) Recursion requires a little brother to do any smaller instance of the problem
(c) You can fix something better than you can fix nothing.
(d) It is inefficient to ask for things you don't want.

Short Answers

1. (32 points) Consider the following ADTs (abstract data types):
   • Binary Search Tree
   • General Tree
   • Hash Table
   • Stack
   • Graph
   • Disjoint Set
   • Priority Queue

   For each scenario below, tell which of the above ADTs you would use to solve the problem.
   You may not use all of the given ADTs, and you may use some more than once. There may be
   more than one correct answer, so clearly justify your choice. (1-2 sentences each)
   a. You are an aspiring chef, and you want to write a program to organize all of your
      recipes. You want to be able to insert, delete, and search for recipes based on their title.
b. You want to write a program that keeps track of your "to-do" list. You want to be able to input tasks and their importance and the program should tell you which job on your list is most pressing.

c. You are about to graduate from college and want to write a program to keep track of your friends' contact information. You should be able to enter a friend's name, phone, and address. You should be able to search for a friend based on her name, and you also want to be able to print out an alphabetical listing of everyone.

d. You are a boss of a branch office, and you want to write a program to keep track of everyone based upon their perceived value to the company. Your program should keep track of everyone and their ranking. Then, when it is time to fire someone, your program should display the person who is least valuable. Sometimes instead of firing someone from a given branch, you need to fire the least valuable person from various branches (treated as one).

e. You are a sophomore and decide to be a computer science/creative writing double major. You will need to plan very carefully to ensure you can complete all of your courses in your remaining three years of school. You have 12 required courses for each major, and some courses must be taken before others. You want to write a program to display a possible ordering of classes such that prerequisites are not lost.

f. You are working for a cable company. You need to suggest a plan for laying the cable such that total costs are minimized and every house had access to cable.

g. You are a school counselor. You need to get the entire German class to Berlin. Not all routes can take all students. You need a plan for getting as many students as possible to Berlin.

h. You work for UPS and need to devise a routing system so that if one location is down, packages can still be delivered to other locations.

2. (20 points) The union/find problem is defined as follows. Suppose we have max objects and are given pairwise equivalence relationships (unions) between objects. Union takes two indices and combines them into the same group. Find takes an index and returns the group name. The following code is an attempted implementation of the union/find problem.

a. Correct any logical errors in the code.

b. Describe changes to union that would make subsequent finds more efficient. Do not consider path compression, as it would be a change to find.

c. Rewrite union so that subsequent finds are more efficient.

```c
int parent [100];

void union( int i, int j )
{
    int group;
```
group = find(j);
    parent[i] = group;
} // union

int find( int i )
{
    int f;
    f = i;
    while ( parent[f] != 0 )
    {
        f = parent[f];
    }
    return f;
} // find
3. (10 points) The min element is deleted from the binomial queue below. Show the result. Make sure you show all steps in the process.

4. (10 points) A tree stored as left-most-child/next-right-sibling is supposed to represent a min heap. Write the code to return true if it is a min heap and false otherwise.

5. Explain the definition of a strongly connected component and its importance.

6. For the B-tree below (with L=5 and M=5), show the tree after inserting 1,5,7,43,47.

7. Show the results of merging the following two leftist heaps.
8. (6 points) Show the results of merging the following two skew heaps if we use the right subtree of the tree with smaller root to merge with the other tree and then swap children of the tree with the smaller root.

![Diagram of two skew heaps](image)