Final Exam Question Bank

1. What is the worst-case time complexity of the quicksort algorithm?
   a) O(n^2)
   b) O(n)
   c) O(n log n)
   d) O(log n)

2. Which of the following is not a quadratic sort?
   a) Bubble Sort
   b) Insertion Sort
   c) Merge Sort
   d) Selection Sort

3. What is the time complexity of binary search in a sorted array?
   a) O(n)
   b) O(log n)
   c) O(n log n)
   d) O(n^2)

4. In a complete binary tree, which traversal method visits all nodes at the same level before moving to the next level?
   a) In-order traversal
   b) Pre-order traversal
   c) Post-order traversal
   d) Breadth-First Search (BFS)

5. What is the worst-case time complexity of merge sort?
   a) O(n^2)
   b) O(n log n)
   c) O(log n)
   d) O(n)

6. Which of the following is not true about recursion?
   a) It is a method of solving problems by breaking them down into smaller subproblems.
   b) Recursion always has a faster runtime than iteration.

c) It requires a base case to terminate the recursion.
d) Each recursive call adds a new frame to the call stack.

7. The height of a balanced binary search tree with n nodes is:
   a) O(log n)
   b) O(n)
   c) O(n log n)
   d) O(n^2)

8. Which of the following is true about the big-O notation?
   a) It provides an exact count of the number of operations an algorithm will perform.
   b) It only describes the worst-case runtime of an algorithm.
   c) It describes an upper bound on an algorithm's growth rate.
   d) It describes the exact time complexity of an algorithm.

9. In a max-heap data structure, which of the following statements is true?
   a) Every node is greater than or equal to its children.
   b) Every node is less than or equal to its children.
   c) The data structure is sorted in increasing order.
   d) The data structure is sorted in decreasing order.

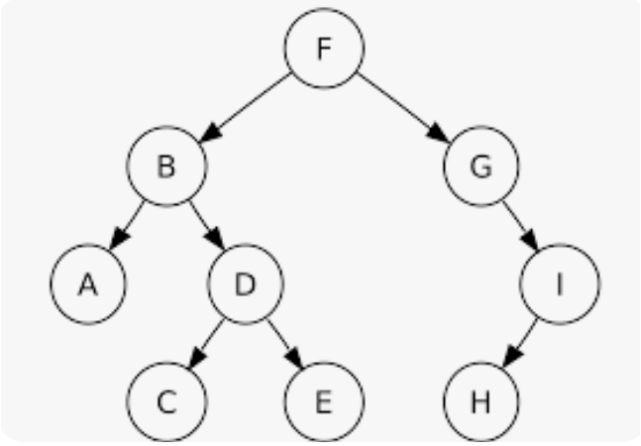
10. Which of the following sorting algorithms have a worst-case time complexity of O(n^2)?
    a) Bubble Sort
    b) Merge Sort
    c) Quick Sort
    d) Insertion Sort
    e) Heap Sort

11. Which of the following statements are true about min-heaps and max-heaps?
    a) Min-heaps and max-heaps are both complete binary trees.
    b) In a min-heap, the root element is always the smallest.
    c) In a max-heap, the root element is always the biggest.
    d) Min-heaps and max-heaps can be used to implement a priority queue.
    e) Extracting the minimum element from a min-heap takes O(logn) time complexity.

12. Which of the following statements are true about proof techniques such as proof by induction, proof by cases, and proof by contradiction?
    a) Proof by cases involves considering all possible cases to show that a statement holds true for each case.
    b) Proof by contradiction is a technique that shows a statement is true by assuming its opposite is true and deriving a contradiction.
    c) Proof by induction is useful for proving properties of algorithms that involve loops or recursion.
    d) Proof by the case is often employed when a problem can be broken down into several distinct scenarios or conditions.

13. You wish to store N elements in one of the two possible data structures: (1) Increasingly Sorted Array (2) Max-heap. For each of the access patterns below, please state which of these two data structures will be best suited. Explain your answer.
    a. Finding the minimum element quickly
    b. Insert an element quickly
    c. Searching for a specific element in the data structure

14. For the below tree, what is the postorder traversal?



15. Define the structure of a binary tree.

struct tree{

// write your code here

};

16. What is the maximum number of nodes at level n in a binary tree?

17. What is the time complexity of searching for a key in a binary search tree?

18. Write a function to find the height of the tree.

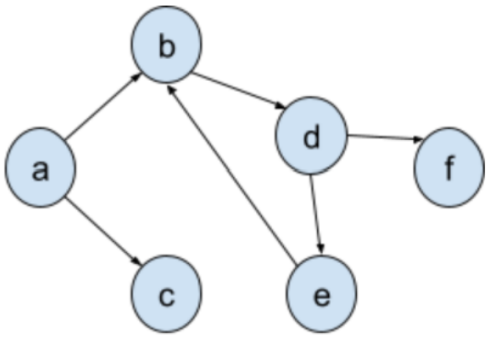
The tree structure is:

```
struct Node {
   int val;
   struct Node* left;
   struct Node* right;
};
```

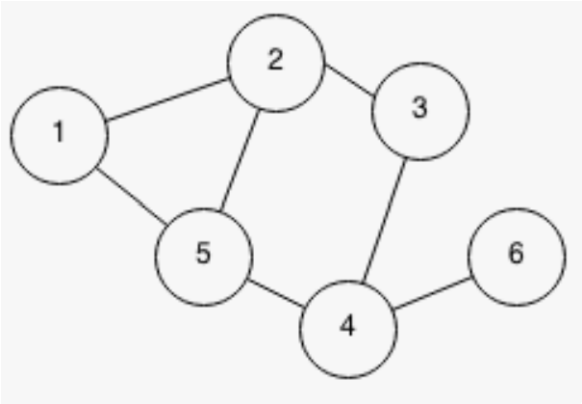
The skeleton of the function is given below. It takes the root node as the parameter.

```
int height(struct Node* root) {
        //write your code here
  }
```

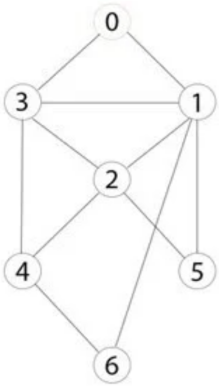
19. Name the cycle in this graph



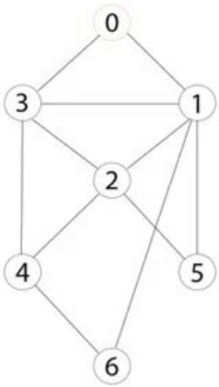
20. Write the adjacency matrix for the following graph.
Also looking at the adjacency matrix, what is the degree of every node?



21. What is the BFS output for the given graph? Start from node 0.



22. What are the 2 most common ways to represent a graph? Explain with examples.



Use the graph above for your examples.

23. Give 3 examples of graphs used in day-to-day life.

24. What is a collision in hashing? And how is it dealt with?

25. What is the time complexity of inserting an element in a hash table in the worst case?

a) O(1)
b) O(log n)
c) O(n)
d) O(n log n)

26. What is the average time complexity of searching in a well-implemented hash table?

27. Write a function to implement a hash table without chaining in C. Use linear probing. The skeleton of the code is provided below. A in the integer array and num in the integer which is to be inserted in the array A.

```
void hash ( int[] A, int num){

    //write your code here

}
```

28. What is linear probing, quadratic probing and double hashing?

29. Suppose we have a hash table with a size of 10 and we want to insert the following keys: 12, 22, 32, 42, 52. The hash function is h(key) = key % 10. Using quadratic probing, what will the hash table look like after inserting all values?

30. Which of the following is an example of dynamic programming?

a) Quicksort algorithm
b) Breadth-first search algorithm
c) Knapsack problem
d) Prim's algorithm

31. What is memoization?

32. Write a function in C to solve the Fibonacci sequence using dynamic programming. Use memoization to avoid redundant calculations.

```
int[] memo;   //array for memoization
int fibonacci ( int number )
{
        //enter your recursive code here
}
```

33. Which of the following is a disadvantage of dynamic programming?
    a) It is difficult to implement.
    b) It requires a large amount of memory.
    c) It can only be used for problems with a recursive structure.
    d) It can be slower than other algorithms for small input sizes.

34. What is a bottom-up approach in dynamic programming?

35. Mark each statement with T (True) or F (False)

(A) Quadratic sorting algorithms have a time complexity of O(n^2) in the average scenarios

(B) QuickSort is always faster than MergeSort

(C) In a binary max-heap, you can remove the root node in O(log n) time complexity

(D) In a binary search tree, a pre-order traversal of the nodes will produce a sorted sequence of the node values.

36. Prove that the sum of two odd integers will always be an even integer. (Hint: use contradiction)

37. What is the difference between a binary tree and a binary search tree?

38. What is the time complexity of inserting an element into a binary search tree?

39. What are the key differences between breadth-first search (BFS) and depth-first search (DFS) in tree traversal?

40. What is a heap data structure, and what are its primary use cases?

41. What are the time complexities of the following sorting algorithms: bubble sort, insertion sort, quick sort, merge sort, and heap sort?

42. Implement a C function that sorts an array of integers based on their frequencies within the array. You are free to use any soring algorithms.

You can assume the elements of the array is between 1 and 10

For Example:

[5, 1, 1, 1, 2, 2, 2, 2, 3, 3]

After Sorting:

[5, 3, 3, 1, 1, 1, 2, 2, 2, 2]

Since Frequency:

5:1, 3:2, 1:3, 2:4

The function signature is provided below:

void sortByFrequency(int *array, int arraySize)

43. Here is a C program that performs the QuickSort algorithm on an array of integers.

```
#include <stdio.h>


int partition(int arr[], int low, int high) {
  int pivot = arr[high];
  int i = (low - 1);
  int temp;


  for (int j = low; j <= high - 1; j++) {
    if (___) { // Fill in the blank
```

```
            i++;
            temp = arr[i];
            arr[i] = arr[j];
            arr[j] = temp;
        }
    }

    temp = arr[i + 1];
    arr[i + 1] = arr[high];
    arr[high] = temp;

    return (i + 1);
}

void quick_sort(int arr[], int low, int high) {
    if (___) { // Fill in the blank
        int pi = partition(arr, low, high);

        quick_sort(arr, low, ___); // Fill in the blank
        quick_sort(arr, ___, high); // Fill in the blank
    }
}
```

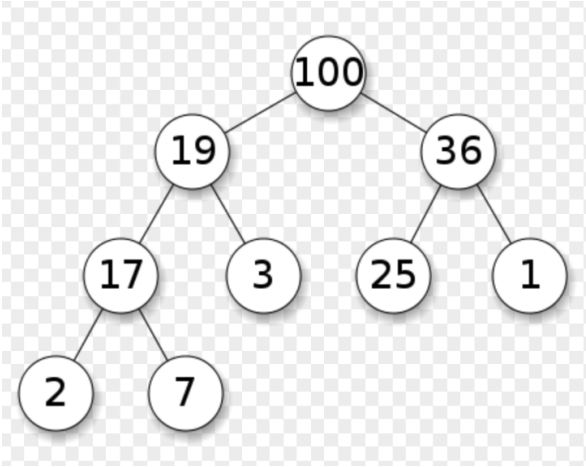
```
int main() {
   int arr[] = {64, 34, 25, 12, 22, 11, 90};
   int n = sizeof(arr) / sizeof(arr[0]);

   quick_sort(arr, 0, __); // Fill in the blank

   printf("Sorted array is: \n");
   for (int i = 0; i < n; i++) {
       printf("%d ", arr[i]);
  }
   printf("\n");

   return 0;
}
```

Please fill out the blanks above so that the function could be able to sort the integers in non-decreasing order.

44. 

(1) Insert 37 into this max heap

(2) Remove the largest node.

draw graphs that illustrate the process of what the tree looks like in each step

45. Given an array that the elements are sorted in ascending order.

Implement a recursive function that converts this tree into a Binary Search Tree

```
/**
* Definition for a binary tree node.
* struct TreeNode {
*     int val;
*     struct TreeNode *left;
*     struct TreeNode *right;
* };
*/
struct TreeNode *sortedArrayToBST(int* nums, int start, int end) {
struct TreeNode *node;
if (start > end)
    return NULL;
 root = malloc(sizeof(struct TreeNode));
// add your code here


 return root
```

```
}

/*

the function will be called as such:

int nums[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

struct TreeNode* root = sortedArrayToBST(nums, 0, 9)

*/
```

46. Which of the following is a valid data type in C?
    a) char
    b) float
    c) boolean
    d) string

47. Which of the following is used to declare a function in C?
    a) int myFunction(int a, int b);
    b) myFunction(int a, int b) int;
    c) myFunction(int a, b);
    d) function myFunction(int a, int b);

48. What is the output of the C program:

```
1 int main()
2 {
3    while(true)
4    {
5       printf("RABBIT");
6       break;
7    }
8    return 0;
```

9 }

a) RABBIT
b) RABBIT is printed unlimited number of times.
c) No output
d) Compiler error.

49. What is a struct in C?
a) A type of loop used to execute a block of code as long as the condition is true
b) A way to declare a variable
c) A block of code that performs a specific task
d) A user-defined data type that groups together variables of different data types

50. Which of the following statements about pointers in C is FALSE?
a) A pointer is a variable that holds the memory address of another variable.
b) Pointers can be used to pass variables by reference.
c) The & operator is used to declare a pointer.
d) The * operator is used to dereference a pointer.

51. What will be the value of y if x = 8?

y = (x > 6 ? 4 : 6);
a) Compilation Error
b) 0
c) 4
d) 6

52. The continue statement cannot be used with _____
a) for
b) while
c) do while
d) switch

53. What will be the output of the following code?
#include <stdio.h>
int main()

```
{
   int i = 0, j = 0;
   while (i<5 & j<10)
   {
      i++;
      j++;
   }
   printf("%d %d", i, j);
}
```

a) 5 5
b) syntax error
c) 4 4
d) 10 10

54. What will be the output of the following code snippet?

```
#include <stdio.h>
void solve() {
   int a[] = {1, 2, 3, 4, 5};
   int sum = 0;
   for(int i = 0; i < 5; i++) {
     if(i % 2 == 0) {
        sum += *(a + i);
       }
       else {
          sum -= *(a + i);
       }
   }
   printf("%d", sum);
}
int main() {
         solve();
         return 0;
}
```

a) 2

b) 15
c) Syntax Error
d) 3

55. What is the output of the following code?

```
int arr[5] = {1, 2, 3, 4, 5};
printf("%d", *(arr + 2));
```

A) 2
B) 3
C) 4
D) 5

56. What is the output of the following code?

```
int stack[5];
int top = -1;
stack[++top] = 1;
stack[++top] = 2;
printf("%d", stack[top--]);
```

A) 1
B) 2
C) -1
D) Compilation error

57. Which of the following statements is true about a queue?
A) It uses a "last in, first out" (LIFO) approach.
B) It can only be traversed in one direction.
C) It consists of a series of nodes, each containing both data and a pointer to the next node in the list.
D) It uses a "first in, first out" (FIFO) approach.

58. Which of the following is a valid way to initialize an array in C?
A) int arr[5] = {1, 2, 3, 4, 5};
B) int arr[] = {1, 2, 3, 4, 5};
C) int arr[5] = {1};
D) All of the above

59. scanf() is a predefined function in_____header file.

A) stdlib.h
B) ctype.h
C) stdio.h
D) stdarg.h

60. How many byte(s) does a char type take in C?

A)1
B)2
C)3
D)4