Data Structure and Algorithms

Midterm Practice Questions

1. What is the main advantage that arrays have over linked lists, give an example to support your answer?

- Arrays are faster when it comes to random access, while in linked list random access is not allowed we have to access the elements sequentially. For example with random access we can use binary search to find an element in a sorted array with O(log n), but in a linked list we can not do that.

2. What is Abstract Data Types?

- It is a mathematical and logical model for storing data types, where a data type is defined by its behavior (function) from the point of view of the user (who is using the ADT)

3. Give a brief description of (a) inserting and (b) deleting an element from an array?

- After locating the, insertion index all elements from the insertion index and to the end of the array are shifted one index down in the array.
- After locating the deletion index, we remove the element by shifting all elements below the deletion index one position up in the array.

4. Discuss whether a stack or a queue is the appropriate structure for determining the order in which elements are processed in each of the following situations?
    a. Batch computer programs are submitted to the computer center.
    - First Come First Served (FCFS) is a policy that is accepted for implementing fairness in a computer center. Therefore it is appropriate to use a queue in this case.
    b. Program A calls subprogram B which calls subprogram C, and so on.
    - Since on every function call, we will pause the execution of the caller function to switch to the called function, and we always execute the last called function before we switch back to the caller function. It is appropriate to use a stack

5. What is an Algorithm and what are the properties of a good algorithm?

- Is a step by step (unambiguous) instructions to solve a given problem. A good algorithm should have five properties: precision, uniqueness, feasibility, finiteness, generality.

6. What is algorithm analysis?

- Algorithm analysis focuses on estimating the resources mainly the time and memory resources required by the algorithm to solve a given computational problem.

7. What is the goal of algorithm analysis?

- The goal of algorithm analysis is to compare algorithms to determine which algorithm is the most efficient to solve a particular problem.

8. What is the difference between linear data structures and nonlinear data structures.

- The main difference between linear and nonlinear data structures lie in the way they organize data elements. In linear data structures, data elements are organized sequentially. In nonlinear data structures, a data element can be attached to several other data elements to represent specific relationships that exist among them.

9. What is the advantage of doubly linked list over singly linked list?

- A doubly linked list can be traversed in both forward and backward direction. Also, delete operation in a doubly linked list is more efficient if the pointer to the node to be removed is given.

10. Find the worst-case runtime complexity of the following function

```
int Foo(int n, j){
      int sum = 0; // 2 steps or O(1)
      For (int i = 0; i<n; i++){ // 4 steps or O(1)
            if(i>j) // 1 step
                  sum = sum +1; // repeated n times
            else {
                  for(int k = 0; k<n; k++) // 4 steps
                        sum = sum -1; //repeated n times
            }
      }
}
```

$f(n) = 3 + 4n\ (1 + 4n) = O(n^2)$

11. Convert the following expressions from infix to postfix

A + B * (C - D) / (E + F)) = **// invalid :)**

(A + B) * C - D = **A B + C * D -**

12. How to design a stack such that GetMinimum(Stack *myStack) takes O(1)?

- Use two stacks: one to store actual stack elements and the second stack to store minimum values. The idea is to do push() and pop() operations in such a way that the top of the second stack is always the minimum.

  For example, let S1 and S2 be our two stacks
  Here are the elements we will insert in into the stack
  5 , 8, 9, 2, 6, 1
  S1 = 5 <<< Top
  S2 = 5 <<< Top
  S1 = 5, 8 <<< Top
  S2 = 5, 5 <<< Top
  S1 = 5, 8, 9 <<< Top
  S2 = 5, 5, 5 <<< Top

S1 = 5, 8, 9, 2 <<< Top
S2 = 5, 5, 5, 2 <<< Top
S1 = 5, 8, 0, 2, 6 <<< Top
S2 = 5, 5,5, 2, 2 <<< Top
S1 = 5, 8, 0, 2, 6, 1 <<< Top
S2 = 5, 5,5, 2, 2, 1 <<< Top

Now as we can see S2 always maintain at the top the smallest elements in S1. Now when we pop an element from the S1 top we also pop an element from S2. To get the minimum value in S1 we simply return the top of S2

Note: There is a more elegant solution. But the one above is OK

13. How to implement a stack which will support following operations in O(1) time complexity?
    a. Push: which add an element to the top of the stack
    b. Pop: which remove an element from the top of the stack
    c. Find Middle: which will return middle element of the stack
    d. Delete Middle: which will remove the middle element from the stack
    - **We should use a linked list (delete an item from a middle of an array is N/2) We need additional pointer called middle that points to the middle and we should use a doubly linked list to be able to efficiently update the middler pointer after push and pop operations.**

14. Write an algorithm to insert an element into a queue. Use sketches or diagrams to illustrate your answer.
- **Refer to the queue lecture for the queue structure and the edge cases:**

15. Write an algorithm to delete an element from the middle of a linked list. Use sketches or diagrams to illustrate your answer.
- **Refer to the linked list lecture for the linked list structure and the edge cases:**

16. Find the upper bound for:
    a. f(n) = 3n+2n +100 — **O(n) with c = 5 and n_0 = 100**
    b. f(n) = $n^4 + 100n^2 + 50$ — **$O(n^4)$ with c = 2 and n_0 =100**
    c. f(n) = n log n + 10n + 5 — **O(n log n) c = 15 and n_0 =5**
17. Find the worst-case runtime complexity of the following function

```
int f3(int n) {
      int sum = 73;  O(1)
      for(int i=0; i < n; i++) { // n times
            // i times where i > = 5 and i moves to n
```

```
                for(int j=i; j >= 5; j--) ( {
                      Sum--;
                }
          }
     return sum;
    }
```

$f(n) = n(n+1)/2 = O(n^2)$

18. Write an algorithm to reverse a singly linked list. What is the big-oh of your algorithm?

- **Hint:** One solution is starting from the head remove element by element and push the elements into a stack data structure, then after pushing all the elements pop all the elements one by one and add insert them from the head of the empty list.

19. Give a big-O estimate for the number of operations (where an operation is an addition or a multiplication) used in this segment of an algorithm.

```
t := 0
for i := 1 to 3  // O(1)
      for j := 1 to 4 // O(1)
            t := t + ij // O(1)
```

The runtime complexity is O(1)

20. How much time does an algorithm take to solve a problem of size n if this algorithm uses $\mathbf{2n^2 + 2^n}$ operations, each requiring $\mathbf{10^{-9}}$ seconds, with these values of n?

    a) 10 b) 20 c) 50 d) 100

    a) $2 * 10^2 + 2^{10}$ nanoseconds

    and so on

21. Write an algorithm based on the binary search for determining the correct position in which to insert a new element in an already sorted list. What is the runtime to insert an element using this algorithm in a sorted array.

    Hint: This is simply the binary search algorithm in addition to shifting the elements in the array to make a space for the new element O(n) = log n + n