BRAC UNIVERSITY
Department of Computer Science and Engineering

Examination: Mid Semester Exam | Semester :Fall 2022
Duration: 1 Hour | Full Marks: 30

CSE 221: Algorithms

Set: A

Answer the following questions.
Figures in the right margin indicate marks.

Name:	ID:	Section:

Total Marks : 10

Jack loves to play with integers. He created a list of n integers where the even indices hold numbers in decreasing order and the odd indices hold numbers in increasing order. For example, this is a list of n=8 integers Jack made.

Index	0	1	2	3	4	5	6	7
Number	23	2	19	3	7	11	5	13

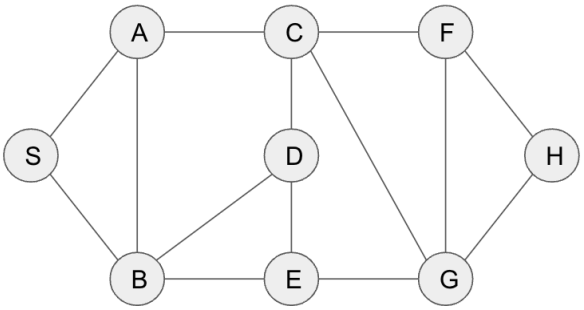
[Explanation:
The indices 1, 3, 5, and 7 have numbers 2, 3, 11, and 13 in increasing order.
The indices 0, 2, 4, and 6 have numbers 23, 19, 7 and 5 in decreasing order.]

You sorted the the list in quadratic time [O(n^2)]. To your utter surprise, Jack replies, "It could be sorted in linear time".

1) ***[CO2]*** **Show** the steps how you sorted the list using a suitable algorithm. Mention the name of the algorithm. — 3
2) ***[CO2]*** **Describe** how Jack sorted the list in linear time. Show the steps too.— 4
3) ***[CO3]*** Jack wants to add the number 15 to your sorted list from (1). He wants you to come up with an efficient idea about how he is going to find the accurate index position for 15 in the list.
   Describe how you are going to assist him in this. **Show** the steps too. — 3

Total Marks: 10**[CO3]**

Consider this graph:



a. **Justify** $\sum_{v \in V} deg(v) = 2m$, where m = number of edges in the graph. 2

b. **Create** an adjacency matrix for this graph.

c. **Apply** Depth First Search to traverse this graph considering S to be the source vertex. **Demonstrate** the results. 2

6

Total Marks:10

a.**[CO7]Calculate** the time complexity of the following function

Finding_Worst_Case(n):

```
int i,j,k,m,multi,a,b,c
for( i = n; i >= 1; i = i / 7 ){
      for( j = 1; j <= n; j = j + 3 ) {
            for( k=1; k<=40 ; k=k+1){
                  multi=a*b
             }
            for( m=n ; m>=1 ; m=m-5 ){
                  multi=multi*c
            }
       }
 }
```

b.**[CO7] Calculate** the time complexity of the following recurrence relation.

[Any method is acceptable as long as steps are shown]

$$T(n) = T(n/2) + T(n/4) + n$$

BRAC UNIVERSITY
Department of Computer Science and Engineering

Examination: Mid Semester Exam — Semester :Fall 2022
Duration: 1 Hour — Full Marks: 30

CSE 221: Algorithms

Set: B

Answer the following questions.
Figures in the right margin indicate marks.

Name:	ID:	Section:

Total Marks : 10

Jack loves to play with integers. He created a list of n integers where the even indices hold numbers in decreasing order and the odd indices hold numbers in increasing order. For example this is a list of n=8 integers Jack made.

Index	0	1	2	3	4	5	6	7
Number	33	7	29	11	17	23	13	31

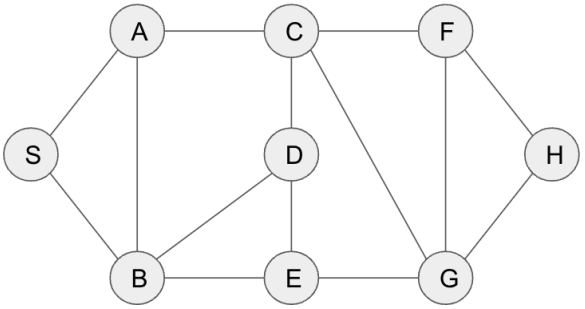
[Explanation:
The indices 1, 3, 5, and 7 have numbers 7, 11, 23, and 31 in increasing order.
The indices 0, 2, 4, and 6 have numbers 33, 29, 17, and 13 in decreasing order.]

You sorted the the list in quadratic time [O(n^2)]. To your utter surprise, Jack replies, "It could be sorted in linear time".

1) ***[CO2]*** **Show** the steps how you sorted the list using a suitable algorithm. Mention the name of the algorithm. — 3
2) ***[CO2]*** **Describe** how Jack sorted the list in linear time. Show the steps too. — 4
3) ***[CO3]*** Jack wants to add the number 12 to your sorted list from (1). He wants you to come up with an efficient idea about how he is going to find the accurate index position for 12 in the list.
   Describe how you are going to assist him in this. **Show** the steps too. — 3

Total Marks:10**[CO3]**

Consider this graph:



a. **Justify** $\sum_{v \in V} deg(v) = 2m$, where m = number of edges in the graph. 2
b. **Create** an adjacency list for this graph.
c. **Apply** Breadth First Search to traverse this graph considering S to be the source vertex. **Demonstrate** the results. 2

6

Total marks:10

a. [**CO7**]**Calculate** the time complexity of the following function

Finding_Worst_Case(n):

```
int i,j,k,m,multi,a,b,c
for( i = n; i >= 1; i = i - 4 ){
        for( j = 1; j <= n; j = j * 3 ) {
            for( k=1; k<=20 ; k=k+1){
                multi=a*b
            }
            for( m=n ; m>=1 ; m=m/5 ){
                multi=multi*c
            }
        }
 }
```

b.**[CO7] Calculate** the time complexity of the following recurrence relation.

[Any method is acceptable as long as steps are shown]

$$T(n) = T(n/3) + T(n/5) + n$$