Kingdom of Saudi Arabia
Ministry of Education
College of Computer
Department of Computer Science

المملكة العربية السعودية
وزارة التعليم
كلية الحاسب
قسم علوم الحاسب

Course Code and Name: CS211: Concepts of algorithms
MID Exam
461 - Date: 15/10/2024 Duration: 1 hour

Student Name:	**Student ID:**	S.N

For Instructor use only

CLOs	k1	k2	k3	S1	S2	S3	S4	V1	V2
Question number									
Allocated mark	1,2,4				3,5				
Earned mark	13				12				

Total Marks

The exam starts here

Question 1 (4 pts)

Choose the correct answer:

1- Which of the following algorithms has the best worst-case time complexity?
   a. Insertion Sort
   b. Selection Sort
   c. Merge Sort

2- In Merge Sort, what is the time complexity of merging two sorted subarrays?
   a. O(log n)
   b. O(n)
   c. $O(n^2)$
   d. O(n log n)

3- What is the time complexity of Insertion Sort in the worst case?
   a. O(n)
   b. $O(n^2)$
   c. O(log n)
   d. O(n log n)

4- In Selection Sort, how many comparisons are made in the best case for an array of size n?
   a. $O(n^2)$
   b. O(log n)
   c. O(n)
   d. $O(n^3)$

Question 2 (6 pts)

Determine the total cost and the complexity (Big O) of the following codes:

Code	Cost	Times

`for(int i=0; i< s; i++) {`
 	
`for(j=1; j<=m;j++)`
`sum++;`
`printf("Hello World");`
`}`
	C1
	
	C2
	C3
	C4
		S+1
		
		S*(m+1)
		S*m
		S

Total Cost	(C2+C3)*S*m + (C1+C2+C4)*S + C1	
Complexity	O(S*m)	

`for (int i = 0; i < N; i++) {`
`for (int j = 0; j < N; j++) {`
`res[i][j] = 0;`
`for (int k = 0; k < N; k++) {`
`res[i][j] += mat1[i][k] * mat2[k][j];`
`}`
`}`
`}`
	C1
	C2
	C3
	
	C4
	C5
		N+1
		N(N+1)
		N*N
		
		N*N*(N+1)
		N*N*N

Total Cost	$(C5+C4)*N^3+(C4+C3)*N^2+(C1+C2)*N+C1$	
Complexity	$O(N^3)$	

`int i = 0;`
`int k=1`
`while (i < n) {`
`printf("%d ", k);`
`i++;`
`k *= 2;`
`}`
	C1
	C2
	C3
	C4
	
	C5
	C6
		1
		1
		N+1
		N
		
		N
		N

Total Cost	(C3+C4+C5+C6)*N+C1+C2+C3	
Complexity	O(N)	

Question 3 (8 pts)

Apply **selection sort** and **insertion sort**, respectively, to the following sequence:**18, 7, 150, 2,35,8**. In each pass give the number of comparisons.

Selection sort:

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	Number of comparisons
18	7	150	2	35	8	5
2	7	150	18	35	8	4
2	7	150	18	35	8	3
2	7	8	18	35	150	2

2	7	8	18	35	150	**1**
2	7	8	18	**35**	150	**Sorted**

Insertion sort:

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	Number of comparisons
18	**7**	150	2	35	8	**1**
7	18	**150**	2	35	8	**1**
7	18	150	**2**	35	8	**3**
2	7	18	150	**35**	8	**2**
2	7	18	35	150	**8**	**4**
2	7	8	18	35	150	Sorted

Question 4 (3 pts)

If an algorithm takes 2 seconds to run with the problem size 4, what is the time requirement (approximately) for that algorithm with the problem size 8, if its order is:

O(1) ➔ T(n) = **2 seconds**

O(n log$_2$n) ➔ T(n) = **2*8*log(8)/(4*log(4)) = 6 seconds**

O(n^2) ➔ T(n) = **(2*8^2) / 4^2= 8 seconds**

Question 5 (4 pts)

Provide the recursive version for the following iterative code that prints odd numbers from 1 to N:

Iterative

```
void printOddNumbers(int n) {
    for (int i = 1; i <= n; i++) {
        if (i % 2 != 0) {
            printf("%d ", i);
        }
    }
}
```

Recursive

```
void printOddNumbers(int n) {
    if (n < 1) {
        return;
    }

    printOddNumbers(n - 1);  // recursive call

    if (n % 2 != 0) {
        printf("%d ", n);
    }
}
```