urse: CS 211 "Concepts of Algorithms"
mester 462

Midterm Exam
Duration: 1 hour

Mark / 25

Name:

Student ID:

For Instructor use only

Outcomes	1	2	3	4	5	6
Course Outcomes covering	√	√				

Question	1	2	3	4
Covered Outcome	1	1	2	2
Allocated mark	5	10	4	6
Earned mark				

Total		
1	2	exam
15	10	25

Question1 Select the most suitable option: **[5 Marks]**

1. In a recursive function, each time the function is called, the input should be:

a. Increased away from the base case
b. Close to the base case
c. Randomly generated
d. Exactly the same as the previous call

2. What is the time complexity of Merge Sort in the worst case?

a. O(N)
b. O(N log N)
c. $O(N^2)$
d. O(log N)

3. Which sorting algorithm is best suited for sorted data?

a. Merge Sort
b. Selection Sort
c. Insertion Sort

4. In Merge Sort, how many subarrays are created in each recursive call?

a. 1
b. 2
c. 3
d. Depends on input

5. In Selection Sort, how many comparisons are made in the second pass for an array of size N?

a. N
b. N-1
c. N-2
d. log N

uestion2 [2*5=10 marks]

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

Code	Cost	Times

for(int i=0; i<m ; i++) {
j=1;
while (j <= n) {
j = j * 2;
}
}
	C1
	
	C2
	C3
	C4
		m+1
		
		m
		m(log(n)+2)
		m*(log(n)+1)

Total Cost	m(C3+C4)log(n)+m(C1+C2+2C3+C4)+C1	
Complexity	O(m*log(n))	

int a = 0, b = 0,c = 0;
for (i = 1; i <= n; i++) {
a = a + 1;
}
for (j = 1; j < m; j++) {
b = b - 1;
}
for (j = 1; j < s; j++) {
c = c - 1;
}
	C1
	C2
	C3
	
	C4
	C5
	
	C6
	C7
		1
		n+1
		n
		
		m
		m-1
		
		s
		s-1

Total Cost	(c2+c3)n+(c4+c5)m+(c6+c7)s+(c1+c2-c5-c7)	
Complexity	O(n+m+s)	

for(int i=0; i< s; i++) {
if(value<0)
write("small value")
else {
for(j=1; j<=m;j++)
sum++
}
}
	C1
	C2
	C3
	C4
	C5
		S+1
		S
		S
		S*(m+1)
		S*m

Total Cost	<= s*(c1+c2)+s*max(c3, m*(c4+c5)+c4)+c1	
Complexity	O(S*m)	
for (int i = 0; i < N; i++) {	C1	N+1

`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];`
`}`
`}`
`}`
	C2
	C3
	C4
	C5
		N*(N+1)
		N*N
		N*N*(N+1)
		N*N*N

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

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

Total Cost	(c2+c3+c4) Log(n)+C1+C2	
Complexity	O(log(n))	

Question3: [3+1=4 Marks]

If an algorithm takes 4 seconds to run with the problem size 16, what is the time requirement (approximately) for that algorithm with the problem size 32?

1. If its order is:

   O(1) ➔ T(n) = 4

   $O(\log_2 n)$ ➔ $T(n) = 4*(\log_2 32/ \log_2 16)=5$

   $O(n^3)$ ➔ $T(n) = 4*(32^3/16^3)=32$

2. Which time complexity from the ones listed above is the most efficient? **O(1)**

Question 4: [6 Marks].

Apply the *Insertion sort* on the following sequence: 95, 70, 100, 40, 90, 20. In each pass give the number of comparisons.

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	Number of comparisons
95	70	100	40	90	20	1
70	95	100	40	90	20	1
70	95	100	40	90	20	3
40	70	95	100	90	20	3
40	70	90	95	100	20	5
20	40	70	90	95	100	sorted