Name: ID:

Midterm Exam

There are 8 questions and 45 points + 5 bonus points. Time limit : 2 hours

Question 1 [6 Points] Fill in the blanks and prove:

(a) $a + (a + d) + (a + 2d) + \ldots + (a + (n-1)d) =$ ______________

(b) $1/2 + 1/4 + 1/8 + \ldots =$ ______________

(c) $a + ax + ax^2 + \ldots + ax^{(n-1)} =$ ______________

Question 2 [6 Points] Order these functions in the increasing order of complexity. No need to prove.

$\log(n^{1/2})$, 2^n, $n^{1/3}\log n$, $n^{1/4}\log n$, $n\log(n)$, $n^{1/3}$, $\log(n/2)$, $(4/3)^n$

Question 3 [6 Points]

Show all the steps in partitioning the given array using two indices i and j. The pivot is at the last index location.

{7, 2, 8, 4, 9, 6, 1, 7, 3, 5}. **The pivot is 5.**

Note: You must use the algorithm explained in the class.

Question 4 [6 Points] → Name of algorithm

(a) *If an array has 100 integers, what is the expected number of inversions?*

(b) Give three sorting algorithms you know that are inversion bound. What are their average time complexities?

(c) Give two sorting algorithms you know that are NOT inversion bound. What are their average case time complexities?

The Master Theorem

For recurrences that arise from Divide-And-Conquer algorithms (like Binary Search), there is a general formula that can be used.

Theorem. Suppose $T(n)$ satisfies

$$T(n) = \begin{cases} d & \text{if } n = 1 \\ aT(\lceil \frac{n}{b} \rceil) + cn^k & \text{otherwise} \end{cases}$$

where k is a nonnegative integer and a, b, c, d are constants with $a > 0, b > 1, c > 0, d \geq 0$. Then

$$T(n) = \begin{cases} \Theta(n^k) & \text{if } a < b^k \\ \Theta(n^k \log n) & \text{if } a = b^k \\ \Theta(n^{\log_b a}) & \text{if } a > b^k \end{cases}$$

Question 5 [6 Points]

Dr. MT was designing an algorithm. He found that he can solve the problem by DAC (Divide And Conquer). He has the following two options:

Algorithm A: Divide the problem into 7 sub problems each of size n/3. Further there is only an O(1) cost in combining the solutions of those 7 sub-problems to obtain the solution of the problem.

Algorithm B: Divide the problem into 4 sub problems each of size n/4. Further, there is an O(n) cost in combining the solutions of those 4 sub-problems to obtain the solution of the problem.

(a) What is the time complexity of the Algorithm A?
(b) What is the time complexity of the Algorithm B?
(c) Which is the best option as far as time complexity is concerned? Justify your answer

Question 6 [6 Points]

A bucket has 5 Red, 3 White and 2 Blue balls. Miss Jane Austin was picking a ball at random. After observing the color, she puts it back in the bucket. Answer next three questions based on this fact.

(a) On an average, how many times Miss Jane has to pick the ball to get a Blue ball?

(b) On an average, how many times Miss Jane has to pick the ball to get a 10 Blue balls?

(c) On an average, how many Red balls Miss Jane has to pick before she can get 10 Blue balls?

Question 7 [6 Points]

(This could be the easiest or the most difficult problem. So, if you are not sure, try at the end)

(a) What is $1^3 + 2^3 + 3^3 + \ldots + n^3$?

(b) Prove your "Guess" using mathematical induction.

Q1. a. $(2a + nd - d)n = \frac{2an + n^2 d - nd}{2}$ ✓

b. $s = 1/2 + 1/4 + 1/8 + \cdots$

$- \; s/2 = \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \cdots$ $\Rightarrow$ $s - s/2 = \frac{1}{2}$ $\frac{1}{2}s = \frac{1}{2}$

$s = 1$ ✓

c. $s = a + ax + ax^2 + \cdots + \cancel{ax^{(n-1)}}$

$- \; sx = \cancel{ax} + \cancel{ax^2} + \cdots \cancel{ax^{(n-1)}} + ax^n$ ✓

$s - sx = a - ax^n \Rightarrow s = \frac{a - ax^n}{1 - x}$

Q2. $\log(n^{1/2})$, $\log(n/2)$, $n^{1/4} \cdot \log n$, $n^{1/3}$, $n^{1/3} \log n$, $n \log(n)$, $\left(\frac{4}{3}\right)^n$, 2^n

Q3.

0	1	2	3	4	5	6	7	8	9
7	2	8	4	9	6	1	7	3	5
i								j	

swap

3	2	8	4	9	6	1	7	7	5
		i				j			

swap

3	2	1	4	9	6	8	7	7	5
			j	i					

ends here now swap i with pivot

3	2	1	4	5	6	8	7	7	9

L E G

4. a. $\frac{1+2+\cdots+99}{2} = \frac{100\cdot 99}{4} = 2475$

b. Insertion Sort, Selection sort, Bubble sort => all $O(n^2)$

c. Bubble sort = $O(n+k)$ k. is size of Bucket
Raddix Sort = $O(n)$

Q5. Algo A => $T(n) = 7\cdot\left(\frac{n-1}{3}\right) + c\cdot 1$
Algo B => $T(n) = 4\left(\frac{n-1}{4}\right) + c\,n$

a. $a=7$ $b=3$ $k=0$ >1 $\Theta(n^{\log_3 7})$

b. $a=4$ $b=4$ $k=1$ $4=4$ $\Theta(n\cdot\log n)$

c. $\Theta(n^{\log_3 7}) > \Theta(n\cdot\log n)$ because $1 < \log_3 7 < 2$ [1, 2]

Q6. a. pick blue ball $P = \frac{2}{10} = \frac{1}{5}$ $\frac{1}{\frac{1}{5}} = 5$

b. $5\cdot 10 = 50$

c. $50-10=40$ $40\cdot\frac{5}{8} = 25$ Red Ball.

$1^3 + 2^3 + 3^3 + \cdots + n^3 =$

step 1. $n=1$ $\quad 1^3 = 1 \quad = 1^2$

step 2 $\quad n=2 \quad 1^3+2^3 = 9. = 3^2$

step 3 $\quad n=3 \quad 1^3+2^3+3^3 = 36 = 6^2$

step 4: assume $n=k \quad 1^3+2^3+3^3+\cdots k^3 = \left(\frac{(1+k)\cdot k}{2}\right)^2$ ✓

step 5. $n=k+1$

$\underbrace{1^3+2^3+3^3+\cdots k^3}+(k+1)^3 = \left(\frac{(k+2)(k+1)}{2}\right)^2 \Rightarrow \frac{k^4+6k^3+13k^2+12k+4}{4}.$ lets prove

$\left(\frac{(1+k)k}{2}\right)^2 + (k+1)^3 = \left(\frac{k^2+k}{2}\right)^2 + k^3+3k^2+3k+1 =$

$= \frac{k^4+2k^3+k^2}{4} + \frac{4k^3+12k^2+12k+4}{4} = \frac{k^4+6k^3+13k^2+12k+4}{4}$

Q.8. a. 4K because tripling size of array takes 3k plus copying elements. so 4K

b. minimum integer → 7

c.

item #	operation	cost of us	customer paid	profit	Balance.
				6	6
1	add	1slot 1add	7		2
2	add	4 to resize 1add 3slot	7	6.	8.
3.	add	1add. 3slot	7	6	14.
4.	add	12 to resize 9slot 1add	6 7	6..	2 8
5	add	1add 9slot	7	6	14
6	add	1add 9slot	7	6	20.
7	add	1add 9slot	7	6	26

item	operation	cost trus	customer paid	profit	Balance
8	add	1 add 9 slot	7	6	32
9	add	1 add 9 slot	7	6	38
10	add	36 for resize 27 slot 1 add	7	6	2 8

Question 8[8 Points] Based on the Lab W1D5. Array resize problem.

c function: 1 to add.

3k to resize (if k >0. Note: k is the size of the "completely filled array")

ĉ function: 7 to add. (customer is willing to pay for add)

0 to resize (customer do not want to pay for resizing. It is not his/her concern)

Item #	Operation	Cost for us	Customer paid	Profit	Balance
1	Add	We assume we start with 1 slot. We add 1 item at the cost of 1.	7	6	6
2	Add	3 to resize (We have two slots) 1 to add	7	6	3 9
3	Add	6 to resize (We have 4 slots) 1 to add	7	6	3 9
4	Add	1 to add	7	6	15
5	Add	12 to resize (We have 8 slots) 1 to add	7	6	3 9
6	Add	1 to add	7	6	15

2

Let us make a small change. Instead of "doubling" the size during the array resize operation, you are "tripling" the size during the resize operation.

Please show the changes required. In particular,

(a) What is c(resize) ? **You must justify.** (Previously it was 3k)

(b) What is the **minimum integer** value possible for ĉ(add). Hint: The balance value must be 0 or positive for any number of add operations.(Previously 7)

Keep c(add) as 1 and ĉ(resize) as 0. **(Previous values. No change)**

(c) Create a table similar to the one shown with "new values". **Show 10 rows.**

Have a nice weekend!

Qn.	1	2	3	4	5	6	7	8	Total
Max.	6	6	6	6	6	6	6	8	50
	6	6	6	6	6	6	6	8	50