PDPM-IIITDM, JABALPUR
MID SEMESTER EXAMINATION
BTech IV Semester (Session 2023-24)

Name of the subject: Design & Analysis of Algorithms (CS2007) Roll No.

Time: 02 Hours Maximum Marks: 20

Q1. a) **[1 Marks]** Arrange the following functions in increasing order of asymptotic complexity.

(i) $f1(n) = 2^n$ (ii) $f2(n) = n^{(3/2)}$ (iii) $f3(n) = n*\log(n)$ (iv) $f4(n) = n^{\log(n)}$

b) **[2 Marks]** Consider the following code segment and determine the growth of f(x) as a function of x.

int f (int x)
{ if (x < 1) return 1;
else return (f(x-1) + g(x))
}
	int g (int x)
	{ if (x < 2) return 2;
	else return (f(x-1) + g(x/2));
	}

c) **[1.5 Marks +1 Marks]** Formulate the recurrence relation for the provided code segment and apply the master theorem for its solution. Additionally, determine the output of the code.

void crazy(int n, int a, int b)
{ if (n <= 0) return;
crazy(n - 1, a, b + n);
printf("%d %d %d n",n,a,b);
crazy(n-1, b, a+n); }
	int main()
	{
	crazy(3, 4, 5);
	return 0;
	}

Q2. a) **[1 Marks +0.5 Marks]** Pan has discovered a way of multiplying *68×68* matrices using *132464* multiplications, a way of multiplying *70×70* matrices using *143640* multiplications, and a way of multiplying *72×72~~72×72~~* matrices using *155424* multiplications. Which method yields the best asymptotic running time when used in a divide-and-conquer matrix-multiplication algorithm? How does it compare to Strassen's algorithm?

b) **[2 Marks]** Use a recursion tree to give an asymptotically tight solution to the recurrence $T(n) = T(\alpha n) + T((1 - \alpha)n) + cn$, where α is a constant in the range $0 < \alpha < 1$ and $c > 0$ is also a constant.

c) **[2 Marks +0.5 Marks]** Usually insert and extract min/ max are allowed operation on max/ min binary heap. It is desired to make one more operation, search, which finds any user inputted element X in a given min or max Binary heap. Propose an efficient scheme and analyze its time complexity.

Q3. a) **[1.5 Marks +0.5 Marks]** You have been n wires of different lengths, with an objective to connect all the n wires into one long wire. The cost to combine any two wires (Wi and Wj) and make them single wires (Wc) is equal to the sum of their (Wi and Wj) lengths.

i. Using efficient data structure(s), propose an efficient scheme to connect the wires with minimum cost.
ii. For the given 10 wires of lengths 14, 16, 13, 18, 12, 7, 15, 4, 1, and 3, compute the minimum cost required to connect all wires to make a single wire.

b) **[3 Marks]** Suppose that you are given *n red* and *n blue* water jugs, all of different shapes and sizes. All red jugs hold different amounts of water, as do the blue ones. Moreover, for every red jug, there is a blue jug that holds the same amount of water, and vice versa. Your task is to find a grouping of the jugs into pairs of red and blue jugs that hold the same amount of water. To do so, you may perform the following operation: pick a pair of jugs in which one is red, and one is blue, fill the red jug with water, and then pour the water into the blue jug. This operation will tell you whether the red or the blue jug can hold more water, or that they have the same volume. Assume that such a comparison takes one time unit. Propose an efficient algorithm that makes a minimum number of comparisons to determine the grouping. Remember that you may not directly compare two red jugs or two blue jugs.

c) **[1.5 Marks]** Professor Midas drives an automobile from Newark to Reno along Interstate 80. His car's gas tank, when full, holds enough gas to travel n miles, and his map gives the distances between gas stations on his route. The professor wishes to make as few gas stops as possible along the way. Give an efficient method by which Professor Midas can determine at which gas stations he should stop and prove that the proposed strategy yields an optimal solution.

Q4. **[2 Marks]** Consider the string "*abbccddeee*". Each letter in the string must be assigned a binary code satisfying the following properties: For any two letters, the code assigned to one letter must not be a prefix of the code assigned to the other letter. For any two letters of the same frequency, the letter which occurs earlier in the dictionary order is assigned a code whose length is at most the length of the code assigned to the other letter. What is the minimum length of the encoded string?

PDPM-IIITDM, JABALPUR
END SEMESTER EXAMINATION
BTech IV Semester (Session 2023-24)

Name of the subject: Design & Analysis of Algorithms (CS2007) Roll No. 22BCS098

Time: 03 Hours Maximum Marks: 30

Q1.

a) **[1.5 Marks]** Consider the weights and values of items listed below. Note that there is only one unit for each item.

Item number	Weight (in Kgs)	Value (in rupees)
1	10	60
2	7	28
3	4	20
4	2	24

The task is to pick a subset of these items such that their total weight is no more than 11 Kgs, and their total value is maximized. Moreover, no item may be split. The total value of items picked by an optimal algorithm is denoted by V_{opt} and the total value of items picked by the greedy algorithm is denoted by V_{greedy}. Find the difference of values obtained by V_{greedy} and V_{opt} and show that the greedy algorithm fails to find the optimal solution for the above problem.

b) **[1.5 Marks]** A doubly linked list is sorted and contains N items. To delete a record using *the delete* operation, a pointer is supplied for the record that needs to be removed. To perform a decrease-key operation, a pointer is given for the record on which the operation is to be performed.
An algorithm A performs the following operation on the list in the following order: *$\theta(N)$ delete, $O(\log N)$ insert, $O(\log N)$ find*, and *$\theta(N)$ decrease-key*. What is the time complexity of all these operations put together? Discuss all possible steps.

c) **[1 Marks]** Prove that the running time of an algorithm is $\Theta(g(n))$ if and only if its worst-case running time is $O(g(n))$ and its best-case running time is $\Omega(g(n))$.

Q2.

a) **[1.5 Marks]** Let T(n) be the recurrence relation defined as follows: $T(0)=1$, $T(1)=2$, and $T(n) = 5T(n-1) - 6T(n-2)$ for $n \geq 2$. Find the asymptotic upper bound for T(n).

b) **[1+1.5 Marks]** Explain how P, NP, NP-completeness, and NP-hardness relate to each other and discuss the implications of proving P = NP or P ≠ NP.

c) **[1.5 Marks]** Consider a divide and conquer algorithm that divides an input of size n into a subproblems, each of size n/b, and the cost of dividing and merging the problems is given by $f(n) = \Theta(n$^$c)$. The Master Theorem classifies this into different cases based on the value of c in relation to log_b(a). Could you explain the intuition and reasoning behind these classifications?

d) **[1.5 Marks]** Let 'S' be an NP-complete problem and 'Q', 'R' are two other problems not known to be in NP. If problem Q is polynomial time reducible to problem S and problem S is polynomial time reducible to problem R then what will be the correct class(s) (P/NP-Completeness/ NP-Hard) for problems Q and R. Give proper justification.

Q3.

a) **[1.5+1 Marks]** Given a text "ababcababcabcabcd" and a pattern "abcabcd", demonstrate step-by-step how the KMP algorithm works to find occurrences of the pattern in the text. Discuss edge cases where the KMP algorithm performs particularly well or poorly compared to other string-matching algorithms (e.g., naive pattern matching).

b) **[1.5 Marks]** Show how to extend Rabin-Karp to look for an m × m pattern among n × n characters. Show all possible steps.

Q4. **[2+2+2+1 Marks]** Being an editor of a newspaper, your job is to put such news articles (NA) on the front page of the newspaper which attracts maximum readers. To accommodate maximum NA on the front page, it is usual practice to break the NA into parts and print one part/partition (P1) at the front page and second part/ partition (P2) at other page. However, the breaking of the NA may reduce the attraction of the readers. Instead of random size of partitions, let us fix the size of the partitions for each of the NA as follows:

(i) P1=100% and P2 = 0%
(ii) P1 =75% and P2 = 25%
(iii) P1 = 50% and P2 = 50%
(iv) P1 = 25% and P2 = 75%

As an example, Table 1 presents the details like, NA id, count of characters in NA, count of attracted readers if P1 = 100%, 75%, 50%, and 25%.

NA id	Character count	Count of readers if			
		P1=100%	P1=75%	P1=50%	P1=25%
1	100	1000	700	600	500
2	200	1500	1200	1100	1000
3	300	6000	5000	2500	2000
4	200	2500	2400	2300	2200
5	400	9000	8500	8000	5000
6	200	1800	1000	800	600

Considering that maximum C characters can be placed at the front page of the newspaper, propose **(a)** Backtracking and **(b)** an efficient greedy based algorithm (Need to write pseudo-code) to place/ print the NAs on the front page which attracts maximum number of readers. Considering C as 1000, find the count of attracted readers using proposed Backtrack and Greedy approaches for the example given in Table. Also, analyze the running time required by your proposed algorithm.

Q5.

a) **[2+2+1 Marks]** A college professor gives several quizzes during the semester, with negative markings. He has become bored of the usual "Best M out of N quizzes" formula to award marks for internal assessment. Instead, each student will be evaluated based on the sum of the best contiguous segment (i.e., no gaps) of marks in the overall sequence of quizzes. However, the student is allowed to drop up to K quizzes before calculating this sum. Suppose a student has scored the following marks in 10 quizzes during the semester.

Quiz	1	2	3	4	5	6	7	8	9	10
Marks	6	−5	3	−7	6	−1	10	−8	−8	8

Without dropping any quizzes, the best segment is *quiz 5-7*, which yields a total of *15 marks*. If the student is allowed to drop up to *2 quizzes* in a segment, the best segment is *quiz 1-7*, which yields a total of *24* marks after dropping quizzes *2* and *4*. If the student is allowed to drop up to *6 quizzes* in a segment, the best total is obtained by taking the entire list and dropping all *5 negative* entries, yielding *33 marks*.

For $1 \leq i < N$, $0 \leq j \leq K$, let $B[i][j]$ denote the maximum sum segment ending at position i with at most j marks dropped.

(i) Write a recursive formula for B[i][j].

(ii) Propose a dynamic programming (DP)-based algorithm that computes the marks of each student that the professor wants to award.

(iii) Describe the space and time complexity of the proposed DP algorithm.

b) **[1.5 Marks]** Consider two strings A = *"abbaccda"* and B = *"abcaa"* consider "x" be length of the longest common subsequence between A and B and "y" be the number of distinct such longest common subsequences between A and B. What will be the value 10x+ 2y?

c) **[1.5 Marks]** You are given a sequence of numbers A = $a_1, a_2, \ldots, a_n$. An exchanged pair in A is a pair (a_i, a_j) such that $i < j$ and $a_i > a_j$. Note that an element a_i can be part of m exchanged pairs, where m is $\leq n-1$, and that the maximal possible number of exchanged pairs in A is $n(n-1)/2$, which is achieved if the array is sorted in descending order. Develop a divide-and-conquer algorithm that counts the number of exchanged pairs in A in O(n log n) time. Argue why your algorithm is correct, and why your algorithm takes O(n log n) time.

---------------END-------------