CSE 2217: Data Structures and Algorithms - II , Summer 2024

Mid Assignment 2

Marks : 20

<table>
<tr><td>1</td><td>Consider the following table that has been generated after applying the Huffman coding algorithm. Use the data from the table to draw the Huffman tree.

<table><tr><th>Character</th><th>Binary Code</th></tr><tr><td>A</td><td>11</td></tr><tr><td>B</td><td>01</td></tr><tr><td>E</td><td>001</td></tr><tr><td>J</td><td>000</td></tr><tr><td>S</td><td>10</td></tr></table></td><td>3</td></tr>
<tr><td>2</td><td>Your vacations have started, and on a fine day, you visit a nearby movie theater. Several movies are being shown there, and as a movie enthusiast, you want to watch as many movies as possible. Moreover, you want to ensure that you watch each movie completely. You observe the schedule board and note the start and end times for different movies. You further realize that it takes about 10 minutes to exit the hall after each movie and move to another hall due to various tasks like standing in a queue, depositing 3D glasses, etc.

Based on the start and end times, determine which movies you will be able to watch completely. Use a greedy strategy to maximize the number of movies you can watch. Clearly show all the steps of your process.

<table><tr><th>SN</th><th>Movie Name</th><th>Start Time</th><th>End Time</th></tr><tr><td>1</td><td>Frozen</td><td>12:30 PM</td><td>02:30 PM</td></tr><tr><td>2</td><td>Batman vs Superman</td><td>02:00 PM</td><td>04:15 PM</td></tr><tr><td>3</td><td>Wonder Woman</td><td>11:00 AM</td><td>01:45 PM</td></tr><tr><td>4</td><td>The Avengers</td><td>01:00 PM</td><td>03:00 PM</td></tr><tr><td>5</td><td>Shazam!</td><td>01:30 PM</td><td>03:30 PM</td></tr><tr><td>6</td><td>Toy Story</td><td>11:30 AM</td><td>01:00 PM</td></tr><tr><td>7</td><td>Guardians of the Galaxy</td><td>12:00 PM</td><td>02:10 PM</td></tr><tr><td>8</td><td>The Incredibles</td><td>10:00 AM</td><td>12:05 PM</td></tr><tr><td>9</td><td>Spider-Man: Homecoming</td><td>10:40 AM</td><td>12:30 PM</td></tr></table></td><td>5</td></tr>
<tr><td>3</td><td>Explain, with a suitable example, why the minimum coin change problem does not have the greedy choice property.</td><td>2</td></tr>
</table>

<table>
<tr><td>4</td><td>You are planning a hiking trip in the mountains and need to bring some water for your team. As you are an environmentalist, you are concerned about the number of waste bottles you are going to produce along the way. There are 3 types of bottled water available in the market: 1L, 4L, and 6L, and your team needs 14 liters of water in total to survive the hike. Using Dynamic Programming, determine how many bottles you should buy from each type so that it fulfills the need of your team and the total number of waste bottles is minimized.</td><td>4</td></tr>
<tr><td>5</td><td>You are a space explorer preparing for a mission to a distant planet. Your spacecraft has limited cargo space, and you need to choose equipment and supplies to bring with you while maximizing the total "Utility" of the items within this space.

<table><tr><th>SN</th><th>Item name</th><th>Size (Cubic meters)</th><th>Utility (Galactic credits)</th></tr><tr><td>1</td><td>Dried Meat</td><td>3</td><td>150</td></tr><tr><td>2</td><td>Bionic Suit</td><td>5</td><td>200</td></tr><tr><td>3</td><td>Lightsaber</td><td>2</td><td>100</td></tr><tr><td>4</td><td>Space Gun</td><td>7</td><td>300</td></tr><tr><td>5</td><td>MRE (Meal-Ready-to-Eat)</td><td>4</td><td>180</td></tr></table>
Your spacecraft's cargo hold can accommodate a maximum volume of 10 cubic meters. Using Dynamic Programming, determine the optimal combination of items to pack that maximizes the total utility without exceeding the volume limit of your cargo hold, providing the names of the items taken and the maximum utility.</td><td>5</td></tr>
<tr><td>6</td><td>What is the time complexity of the 0/1 knapsack problem if we try to solve it using (I) Brute force, and (II) Dynamic programming?</td><td>1</td></tr>
</table>

United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Midterm Exam Total Marks: **30** Summer 2022

Course Code: CSE 2217 Course Title: Data Structure and Algorithms II

Time: 1 hour 45 minutes

Any examinee found adopting unfair means will be expelled from the trimester / program as per UIU disciplinary rules.

There are **FOUR** questions. **Answer all of them**. Show full simulation/tabulations wherever necessary. Figures in the right-hand margin indicate full marks.

1. **(a)** Derive the best-case, and the worst-case running-time equations for the following function $favouriteSum$ and represent using Asymptotic Notation. **[4]**

```
1  bool favouriteSum(int n, int m) {
2      int sum = 0;
3      for(int i=1;i<=n;i++) {
4          for(int j=1;j<=i;j++) {
5              sum = sum + (i+j);
6          }
7      }
8      for(int i=2;i*i<=m;i++) {
9          if(m%i==0)
10             return false;
11     }
12     if(sum%m==0)
13         return true;
14     else
15         return false;
16 }
```

(b) Derive the exact-cost equation for the running-time of the following function and show that the time complexity is $O(n \log_4 n)$. **[4]**

```
1  void Function(int n) {
2      int prod = 1;
3      for(int i=n; i>=1; i--) {
4          for(int j=n; j>=1; j=j/4) {
5              prod = prod * (i+j);
6          }
7      }
8      printf("%d\n", prod);
9  }
```

2. **(a)** Express the time complexity of Maximum-sum Subarray problem when Brute Force method is applied to solve it. Explain why the Divide-and-Conquer approach can improve the complexity. **[2]**

(b) Given an array of integers A = {-2, 3, -2, 4, -1, 2, 1}, find the Maximum-sum continuous Subarray using divide-and-conquer. You must show the recursion tree and clearly mention left, right and crossing sum for each tree node. **[3]**

(c) Suppose we have two sorted sub-arrays: L: 1, 5, 8, 9, 10, 15 and R: 4, 6, 7, 11, 13, 14. Perform the procedure Merge on L and R to find the final sorted array A. Show each step of your answer and the number of comparisons required in each step. **[2]**

3. **(a)** Given the arrival and the departure times (in minutes) of 8 trains for a railway platform, find out the maximum number of trains that can use that platform without any collision, using a greedy algorithm. There must exist at least 10 minutes of safety break between the departure of one train and arrival of the next one. **[2]**

[1000, 1030], [840, 1030], [850, 1040], [1700, 2000], [800, 835], [1300, 1800], [1500, 1650], [1200, 1380]

Explain your strategy very briefly and show detailed calculations. No need to write pseudocode.

(b) "Data encoded using Huffman coding is uniquely decodable"- is the statement true or false? Justify your answer. **[1]**

(c) A document to be transmitted over the internet contains the following characters with their associated frequencies as shown in the following table:

Character	a	e	l	n	o	s	t
Frequency	84	111	54	45	71	57	69

There are a total of 10000 characters in the document.

Use Huffman technique to answer the following questions:

i. Build the Huffman code tree for the message and find the codeword for each character. Encode "stolen" using the codewords. **[3]**

ii. What is the percentage saving if the data is sent with fixed-length code values without compression? **[1]**

4. **(a)** Given an infinite number of coins with denominations {1,2,3,4,7}, find the minimum number of coins required to make an amount of 15. You must show your working in a tabular format. Also state which coins you are using, and how many of them you are using. **[4]**

(b) Describe the 'Overlapping Subproblem' property of dynamic programming using Fibonacci as a reference problem. **[2]**

(c) Explain how dynamic programming improves running time of 0-1 Knapsack problem. **[2]**

United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Midterm Exam Total Marks: **30** Fall 2022

Course Code: CSE 2217 Course Title: Data Structure and Algorithms II

Time: 1 hour 45 minutes

Any examinee found adopting unfair means will be expelled from the trimester / program as per UIU disciplinary rules.

There are **four** questions. **Answer all of them**. Show full simulation/tabulations wherever necessary. Figures in the right-hand margin indicate full marks.

1.	**(a)** Suppose, A problem X of size n can be divided into three subproblems each of size n/4, each of the problem can be solved recursively in time T(n/4) respectively. The cost of dividing the problem and combining the results of the subproblems is O**(**nlogn). **Formulate** the recurrence relation assuming, T(1) = O(1).	**[1.5]**
	(b) Solve the following recurrence equation: T(n) = 3T(n/3)+O(1), where T(1) = O(1).	**[2.5]**
	(c) Given an array of integers **A** = {2, -3, 2, -4, 1, -3, -2}, find the ***Maximum-sum Continuous Subarray*** using divide-and-conquer. You must show the **recursion tree** and clearly mention **left, right and crossing sum** for each tree node.	**[3]**
2.	**(a)** Following items are available in a grocery shop: ➢ 10 kilogram rice grain which costs 800 taka ➢ 10 kilogram salt which costs 890 taka ➢ 8 kilogram saffron powder which costs 2000 taka and ➢ 4 kilogram sugar which costs 500 taka A group of thieves (Thief 1, Thief 2, … Thief M) have come to steal from that shop, **each with a knapsack of capacity 8 kg**. The thieves are entering in serial, *Thief 2* enters after *Thief 1* is done with stealing, *Thief 3* enters after *Thief 2* is done with stealing and so on. *Since each thief wants to maximize his/her profit*, **how many thieves** will be needed in the group to empty the grocery shop and **what are the items** that each of those thieves carry? Show details of the calculation.	**[3]**
	(b) A document to be transmitted over the internet contains the following characters with their associated frequencies as shown in the following table: There are a total of 1000 characters in the document. I. **Build** the Huffman code tree for the message and find the **codeword** for each character. II. Decode "**0110001111**" using the codewords generated in (i).	**[3+1]**
3.	**(a)** Suppose you have computed a **Fibonacci** series using **dynamic programming**. **Justify** the following statements with **an example**: I. **Overlapping Subproblems** property has been satisfied in your computation. II. Dynamic programming gives you a **more efficient** solution than an obvious recursive algorithm.	**[1.5* 2 =3]**

Character	A	B	C	D	F	T	–
Frequency	40	23	8	10	4	12	3

	(b) What is 'Optimal Substructure' property? How does Dynamic Programming differ from Divide-and-Conquer problems in terms of handling subproblems?	**[2]**
	(c) Suppose, CoffeeLand Coffee Shop charges **50 BDT** (Bangladesh Taka) for each cup of small Americano with an additional vat of **3%**. You bought 2 cups of small Americano and gave the cashier **110 taka**. The cashier has got a huge supply of the following types of coins: **1 taka**, **2 taka, and 5 taka** in the cashbox. You don't want to carry many coins, so you asked the cashier to return the change using a **minimum number of coins**. Determine **the number** and **type of coins** the cashier should return in this scenario by applying the **Dynamic Programming** Approach.	**[3]**

4.	**(a)** Derive the best-case and the worst-case running-time equations for the following function ***calculate*** and represent using Asymptotic Notation.
	`void calculate(int n, int p, int A[]){`
	`  int prod = 0;`
	`  for (int i = 1; i<=n; i++){`
	`  for (int j = 1; j <= i*i; j++){`
	`  prod *= pow(i,j);`
	`  }`
	`  }`
	`  for(int m = 2; m <= p;m++){`
	`  if(A[m] < 100 ){`
	`  break;`
	`  }`
	`  prod = prod * A[m];`
	`  }`
	`  cout<<prod<<endl;`
	`}`
		[4]

	(b) Derive the exact-cost equation for the running-time of the following function and show that the time complexity is $O(n \log n \log_5 n)$:
	`void funFunction(int n)`
	`{`
	`  int sum = 0;`
	`  for (int k = 0; k < n; k*=2){`
	`  for (int j = n/2; j <=n; j++){`
	`  for (int i = n; i >=1; i=i/5){`
	`  sum += (i+j+k);`
	`  }`
	`  }`
	`  }`
	`  cout<<sum<<endl;`
	`}`
		[4]