**Advanced Algorithms**

**Exercise for Lecture 12**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name** |  | **Student ID** |  |
| **Problem 1** |  | | |
| **Problem 2** |  | | |
| **Problem 3** |  | | |
| **Total Score** |  | | |
| **Notes** | Deadline: **2023-11-04 24:00**  Submission Format: ‘**Lecture12\_Name\_Student ID.docx**’, and please send to: **[chenlq1997@126.com](mailto:algorithms_23fall@163.com)**.  This assignment is meant to be an evaluation of your **individual** understanding coming into the course and should be completed **without collaboration** or outside help. | | |

**Problem 1.[30 points]**

There is a backpack with a capacity of 55 and 10 items. Now ignoring the actual geometric shape of the item, we believe that as long as the remaining capacity of the backpack is greater than or equal to the volume of the item, it can be loaded into the backpack, and the item can be split proportionally. Each item has two attributes, namely volume w and value p. How can we pack the item back to maximize the total value of the items in the backpack?

|  |  |  |
| --- | --- | --- |
| Item | Value | Volume |
| 1 | 20 | 5 |
| 2 | 24 | 5 |
| 3 | 28 | 10 |
| 4 | 37 | 15 |
| 5 | 40 | 15 |
| 6 | 28 | 10 |
| 7 | 58 | 20 |
| 8 | 11 | 5 |
| 9 | 32 | 10 |
| 10 | 41 | 15 |

**Solution:**

**The value per unit volume of each item is shown as follows:**

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Value | Volume | Value per unit volume |
| 1 | 20 | 5 | 4 |
| 2 | 24 | 5 | 4.8 |
| 3 | 28 | 10 | 2.8 |
| 4 | 37 | 15 | 2.47 |
| 5 | 40 | 15 | 2.67 |
| 6 | 28 | 10 | 2.8 |
| 7 | 58 | 20 | 2.9 |
| 8 | 11 | 5 | 2.2 |
| 9 | 32 | 10 | 3.2 |
| 10 | 41 | 15 | 2.73 |

**So the order of these items by value per unit volume is**

**2 1 9 7 3 6 10 5 4 8**

**So we can pack item 2, item 1, item 9, item 7, item 3 and 50% of item 6.**

**Problem 2.[30 points]**

There exist 7 characters and we have the occurrence frequency of them. Please give the huffman code of each character.You need show each step of them coding process. Left branch of huffman tree will be 0 and right branch of huffman tree will be 1.

G: 4 D: 8 C: 10 E: 11 B: 16 F: 20 A:31

**Solution:**

Step1: G: 4 D: 8 C: 10 E: 11 B: 16 F: 20 A: 31

Step2: C: 10 E: 11 12 B: 16 F: 20 A: 31

G: 4 D: 8

Step3: 12 B: 16 F: 20 21 A: 31

G: 4 D: 8 C: 10 E: 11

Step4: F: 20 21 28 A : 31

C: 10 E: 11 12 B: 16

G: 4 D: 8

Step5: 28 A: 31 41

12 B: 16 F: 20 21

G: 4 D: 8 C: 10 E: 11

Step6: 41 59

F: 20 21 28 A: 31

C: 10 E: 11 12 B: 16

G: 4 D: 8

Step7: 100

41 59

F: 20 21 28 A: 31

C: 10 E: 11 12 B: 16

G: 4 D: 8

So the coding is f:00, C:010, E:011, G:1000, D:1001, B:101, A:11.

**Problem 3.[40 points]**

There exist one person, he is passionate about symmetrical aesthetics. When he places items, he would place them symmetrically. For example, when he places a pair of chopsticks, a bowl and two spoons, he would place them as follows:

**Chopstick1, spoon1, bowl, spoon2, chopstick2**.

3.1 Can this person place a bowl, a spoon, a pair of chopsticks and two boxes of tissues according to his habits? Please explain the reason.

3.2 For a pile of items, if he cannot place them according to his habits, how many items can he place at most? Not all items will be properly placed. For example, for a bowl, a spoon, a pair of chopsticks and two boxes of tissues, he can place at most 5 items as follows

**Chopstick1, a box of tissues, a bowl or a spoon, a box of tissues, chopstick2**

So please provide a pseudocode based on greedy algorithm which can return the maximum number of items that can be placed according to this person’s habit.

For example, for 2 item A, 2 item B, 3 item C, 4 item D, 5 item E, the result would be 15.

**A B C D D E E C or E E E D D C B A**

**Solution:**

**3.1** No, these items cannot be placed according to his habits. Because there exit more than one item whose number is odd.

**3.2** First we count the occurrence time of each item. Then, for each item, if its number is even, then half of them will be places on the left and the other half of them will be placed on the right. If the number is odd, then the half of them will be placed on the left and the other half of them will be placed on the right, and the remaining one will not be placed. At last, we choose one of these left items to be placed in the middle.

Pseudocode

Count[a,…,z] = 0

For i←1 To s.length() Do //count the occurrence times of each item

count[s[i]] = count[s[i]]+1

num = 0 //The number of items that can be placed according to his habit

For i←1 To count.length() Do

num = num+count[i]/2\*2 //

If count[i]%2==1 && num%2==0 Then //for the items whose number is odd, we choose only one kind of these items to be the middle element.

num = num + 1

return num