Timothy Hoang

5/5/19

CMPT435-111

Assignment 12

**Date Assigned: 04/29/2019**

**Due: Midnight 05/05/2019 on iLearn**

**Please read turn-in checklist at the end of this document before you start doing exercises.**

**Section 1: Pen-and-paper Exercises**

1. Suppose we want to make change for N cents, using the least number of coins of denominations {1, 5, 10} cents (Different from the US currency system!). Consider the following greedy strategy: suppose the amount left to change is M, take the largest coin that is no more than M; subtract this coin's value from M, and repeat.

Does this algorithm output an optimal solution? If not, give a counterexample. If yes, prove that this algorithm always outputs an optimal solution (a formal proof as what we have done in the video).

**2 points – Your answer**

**The greedy algorithm is optimal for that specific currency system. However, in another coin system, it may not be optimal.**

**8 points – Counterexample or proof**

**G: a, b, c**

**O: a1, b1, c1 —> a1+1, b1-2 save 1 coins = O1**

**O1 shouldn’t exist**

1. **a == a1 —> compare b and b1**
2. **a > a1**

**a1 = a**

**b1 = b**

**c1 = c**

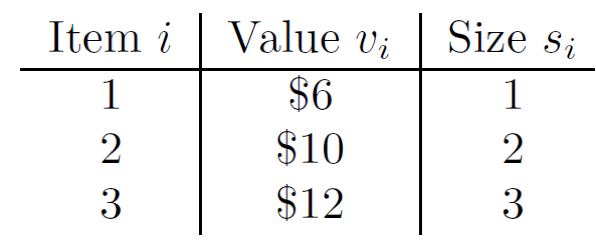
**Optimal doesn’t exist showing that greedy algorithm is optimal**

1. Consider the knapsack problem.

A thief enters a store and sees n items, each item i has value vi, size si. His knapsack is of size S

What items should he put into the knapsack to maximize profit?

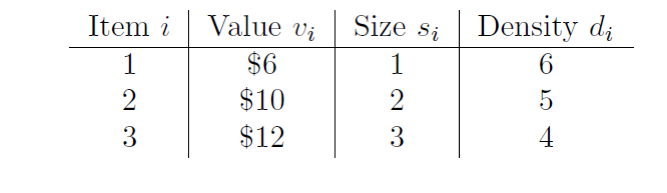
Assume we have n = 3 items, and a bag of size S = 5.



a)

In fractional knapsack problem, the thief can take a fraction of an item and put it into his knapsack. To solve the problem, we can use the following greedy algorithm:

1. Compute di = vi/si for each item
2. Sort each item by their value di.
3. Take as much as possible of the highest di item not already in the bag



Given the algorithm and the items above, which item(s) will you put into the knapsack? What is the total value of your knapsack? (5 points)

Given the algorithm, I would put all of item 1, all of item 2, and 2/3 of item 3. The value of my knapsack is $6 + $10 + ($12\*2/3) = $24

b) In 0-1 knapsack problem, a thief can only choose to take all or nothing of a particular item (hence the 0-1). Given the algorithm and the items above, which item(s) will you put into the knapsack? What is the total value of your knapsack? Is it the maximum value of knapsack? Does the greedy algorithm solve the 0-1 knapsack problem? (5 points)

Given the algorithm, I would take item 1 and item 2 giving my knapsack a value of $16. This is not the maximum value of the knapsack which would be $22 if you took items 2 and 3 so the greedy algorithm does not solve the 0-1 knapsack problem.

1. Given a BST and a positive integer k, find the k\_th smallest element in the BST.

For example, in the following BST, if k = 3, then output should be 10, and if k = 5, then output should be 14.



Design an algorithm to solve this problem.

1. describe the idea behind your algorithm in English (2 points);

Use in-order traverse algorithm to search left, root, right. Every time you scan a number, you move the counter, k, down by one until you hit 0. Once you get 0, that is your answer.

(ii) provide pseudocode (5 points);

Static void inordervisit(Node root){

if (root !=null){

inodervisit(root.left);

visit root;

k—;

if(k == 0){

return root;

}

inordervisit(root.right);

}

if(k == 0){

return root;

}

}

(iii) analyze its running time (3 points).

Olog(n)

**Note: We will discuss this problem in class.**

**Section 2: Java Implementation**

1. Implement the Greedy Coin Changing Algorithm in Java.

Note:

Find a file called Coinchange.java in assignment 12 folder.

Complete the method of greedycoinchange().

Test your method in the main method provided following the comments.

**Important: In all of the assignments of this course, when you are asked to implement an**

**algorithm for a problem, your code will be evaluated based on:**

**5 points - Execution**

**Each file must run without error or warning on valid input described in the main method provided.**

**5 points - Within Code Documentation**

**Is the code documented for obvious understanding of the use, preconditions, and postconditions of each function?**

**20 points - Correctness**

**Is the algorithm implemented correctly? Does your method pass the test?**

1. Implement the problem 3 in Java.

Note:

Find a file called BST.java in assignment 12 folder.

Complete the method of ReturnKthSmallestElement ().

Test your method in the main method provided following the comments.

**TURN-IN CHECKLIST:**

1. **Answers to Section 1 (.doc/.txt/.pdf), and to Section 2 (all your source Code (.java files)). Remember to include your name, the date, and the course number in comments near the beginning of your code/report.**
2. **Create a folder and name it 'FirstName\_LastName\_assignment\_10'. In the newly created folder copy and paste your files (.doc/.txt/.java files). Then compress the folder, and push it to iLearn.**