Challenge Problems in Algorithms and Data Structures

Revised November 28, 2022

The purpose of these problems is to provide a challenge to the top students, meaning those who master the regular material of the course and would like to learn more.

Do not try to work these problem to improve a poor grade. If you are doing badly in the course, concentrate on the regular material.

1. The 13-coin problem is to use a balance scale to find a counterfeit coin among 13 coins, and determine whether it is too heavey or too light. The Wikipedia page Wikipedia page on Balance Puzzles states that there is no algorithm for this problem, but gives no proof. Prove it.

2. Given a linked list of integers, mergesort sort the list in $O(n \log n)$ time without creating any additional nodes. That is, never use the "new" constructor. The resulting sorted list uses exactly the nodes given in the input list, just rearranged.

   Format of the Data. Let $x_1, \ldots x_n$ be the list of integers. The input file has $n + 1$ lines. The first line is the value of n, the length of the list. The $(i + 1)^{\text{st}}$ line contains x_i.

   You must write a solution as a C++ program, which I will test with my own data. You must first read the inputs into a linked list, using the constructor "new" as needed. After that, you may not use "new." Your input variables will consist only of the linked list and a single pointer which points to the first node of the list.

   Your output is a file consisting of n input integers in sorted order.

   I falsely stated in class that this can be done using only constant workspace, where workspace means memory space needed not counting the space holding the linked list. My solution uses $O(\log n)$ workspace.

3. Let G be a weighted acyclic directed graph, with a designated "start" node, s. You wish to find two paths, each starting at s, such that every node of G is on at least one of the two paths. The cost of a solution is the sum of the weights of the edges of the two paths. Design a dynamic programming algorithm that finds a solution of minimum cost, or determines that there is no solution.

   You must write a solution as a C++ program, which I will test with my own data.

4. Give a linear time (that is, $O(n)$ time) algorithm for the Traveler's Problem, given below.

   A traveler must walk from A to his home at B, along a road of length m miles. There are n inns on the road. The i^{th} inn is d_i miles from A and costs c_i to stay at. The traveler can walk at most w miles per day. How can he get home at minimum cost, if he must stay at an inn every night until he gets to B?

   You must write a solution as a C++ program, which I will test with my own data.

Format of the Data. For simplicity, all numbers will be positive integers. The input data is a file consisting of $n+1$ lines of text. The first line contains the values of n and w. For each $i \leq i \leq n$, the $(i+1)^{\text{st}}$ line contains two numbers, d_i and c_i You may assume that $d_i < d_{i+1}$. The n^{th} inn is the traveler's home and is located at B, hence d_n is the distance from A to B, and $c_n = 0$ since he doesn't have to pay to sleep at home.

Your program should state that there is no solution if there is no way the traveler can make the journey. Otherwise, your program should state the minimum cost of a solution.

The time complexity of the obvious dynamic programming algorithm is $O(n^2)$. Can you do better? There is an time $O(n)$ time algorithm for this problem.

Here are some sample runs of my program: the input file followed by the output.

```
5 5
2 3
3 2
6 1
8 4
11 0
```

```
n = 5 w = 5
Walk 3 miles and pay 2
Walk 3 miles and pay 1
Walk 5 miles and pay 0
The traveler is now home.  The total cost of the journey is 3
```

```
 9 5
 3 6
 4 7
 4 8
 9 6
10 3
11 2
14 4
15 8
17 0
```

```
n = 9 w = 5
Walk 4 miles and pay 7
Walk 5 miles and pay 6
Walk 5 miles and pay 4
Walk 3 miles and pay 0
The traveler is now home.  The total cost of the journey is 17
```

```
 9 5
 3 6
 4 7
 4 8
 9 6
10 3
11 2
14 4
20 8
21 0
```

```
n = 9 w = 5
Walk 4 miles and pay 7
Walk 5 miles and pay 6
Walk 5 miles and pay 4
The next inn is too far away for the traveler to walk. There is no solution.
```