
Due on Wednesday 16

Problem 1: We are going on a trip along the Appalachian trail. We have a list of all possible campsites that we can camp in along the way (say n). We want to do this trip in exactly k days, stopping $k - 1$ nights to camp. Our goal is to plan this trip so that we minimize the maximum amount of walking done in a single day. In other words, if our trip involves 3 days of walking, and we walk 11, 14, 12 miles on each day respectively, the cost is 14. Another schedule that involves walking 11, 13, 13 miles on each day has cost 13. The locations of the campsites are specified in advance, and we can only camp at a campsite. Your algorithm should be based on dynamic programming and run efficiently.

Problem 2: You are hired by a financial company, Crazy Buck Ltd., to design a program to find good investments. The company has done a lot of research in n different emerging markets. For market i they have a set of investment strategies $S_{i,1}, S_{i,2}, \dots, S_{i,k}$. Strategy $S_{i,j}$ involves investing $I_{i,j}$ dollars and, according to their research, has a return of $R_{i,j}$. Assume all numbers are integers.

The company has a budget B on how much it can invest. Because two strategies for the same market typically overlap, you can choose at most one strategy per market. The goal is to maximize profit. Show your boss at Crazy Buck how to solve her problem using dynamic programming!

Problem 3: In the knapsack problem we are given a set of n items to fit in a knapsack of size W . Item i has weight w_i and value v_i . We are to find a set $S \subseteq \{1, \dots, n\}$ such that $\sum_{i \in S} w_i \leq W$, which maximizes $\sum_{i \in S} v_i$.

In class we saw a solution for this problem using dynamic programming, here we explore the possibility of using the greedy algorithm on the right. A few natural ways of sorting the items would be in increasing w_i , decreasing v_i , and decreasing $\frac{v_i}{w_i}$. Show that these strategies may fail to find an optimal solution.

GREEDY-KNAPSACK

```
sort items
Let  $S$  be empty
For each item  $i$  [in order]
    If  $w(S) + w_i \leq W$ 
        add  $i$  to  $S$ 
return  $S$ 
```

Problem 4: From the book: Exercise 6.6.

Problem 5: From the book: Exercise 6.13.

Problem (extra credit): Give an $O(n^2)$ time algorithm for problem 1.