CS41 Lab 8

October 28, 2019

The lab this week focuses on dynamic programming. The purpose of this lab is to gain practice using dynamic programming to solve problems. Do **not** expect to solve all problems. Consider lab a success if you solve one of the first two problems and make some progress on another problem.

General Hints:

- Focus on the **choice** you might make to construct an optimal solution.
- In each problem below, you will need to keep track of some extra information. Try to be deliberate about what extra information you might need to keep track of, and why this extra information might be required to efficiently get the dynamic programming to work.
- Initially focus on the first two steps of the dynamic programming process. Don't stress about pseudocode until after you've solved all lab problems.

1. **Making Change With Strange Coins.** In Lab 5, you considered the problem of making change for n cents out of the fewest number of coins. You previously designed a greedy algorithm that gave optimal solutions for US coins, and for coins that are used in a country called Algorithmland. What about for other sets of coin denominations?

   (a) The greedy algorithm from lab 5 (always use the largest denominations possible) will not work for all possible coin denominations. Show that it won't always work by providing a specific n and a specific set of coin denominations $c_1, \ldots, c_k$ for which your greedy algorithm makes change using more coins than necessary.

   (b) Design a dynamic programming algorithm that takes as input the amount of change to make, n, and a list of coin denominations, $c_1, \ldots c_k$. If it is possible to make n cents in change using denominations $c_1, \ldots, c_k$, your algorithm should output the smallest number of coins required. Otherwise, your algorithm should output `FAIL`.

2. **Subset Sum.** In this problem, you are given an integer *weight threshold* $W > 0$ and a list of n items $\{1, \ldots, n\}$ each with nonnegative weight w_i. Your task is to output a subset of items $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} w_i$ is as large as possible, subject to $\sum_{i \in S} w_i \leq W$.

   Design an analyze a dynamic program to solve **Subset Sum**. Your algorithm should run in $O(nW)$ time.

3. **Gerrymandering (K&T 6.24)** *Gerrymandering* is the practice of carving up electoral districts in very careful ways so as to lead to outcomes that favor a particular political party. Recent court challenges to the practice have argued that through this calculated redistricting, large numbers of voters are being effectively (and *intentionally*) disenfranchised.

   Suppose we have a set of n precincts $P_1, \ldots, P_n$, each containing m registered voters. We're supposed to divide these precincts in to two *districts*, each consisting of $n/2$ precincts. Now, for each precinct, we have information on how many voters are registered to each of two political parties. Say that the set of precincts is *susceptible* to gerrymandering if it is possible to perform the division into two districts in such a way that the same party holds a majority in both districts.

Give an algorithm to determine whether a given set of precincts is susceptible to gerrymandering. The running time of your algorithm should be polynomial in n and m.

For example, suppose there are four precincts, and two political parties A and B. Letting A_i and B_i be the number of voters in precinct i of each political party, Suppose we have

$$A_1 = 55, A_2 = 43, A_3 = 60, A_4 = 47 \textit{ and}$$

$$B_1 = 45, B_2 = 57, B_3 = 40, B_4 = 53 \ .$$

This set of precincts is susceptible to gerrymandering since pairing precincts 1 and 4 together and $2, 3$ together gives party A a 102 - 98 majority in the first district and a 103 - 97 majority in the second.

Hints:

(a) The table you store will likely be multi-dimensional. It will be helpful to start by thinking hard about what information you will need to maintain.

(b) Start by looking at the specific example above. How would you build up the assignment of precincts that gives the gerrymandered districts?

(c) In the general case, if precincts are suceptible to gerrymandering, how many majority-party votes can be assigned to district 1? How many to district 2?