CSC 495 – Problem Set 5 – Due Tuesday, February 14

Problem 5.R1: Do Right By Dudley

Required Problem
Points: 50 points
5 extra ninja points if a correct solution is submitted before 5:00PM on Feb 8

Background

In the first Harry Potter book, Dudley Dursley is very upset about the number of presents he received, despite assurances that some of the presents were nicer than the ones he had gotten the year before. Your job is to help poor Uncle Vernon rectify this horrible slight by maximizing the number of presents that Dudley receives. Fortunately, there is a fantastic shop that has everything that Dudley could possibly want, which is in fact where Uncle Vernon bought the initial set of presents. The store is nice enough to accept returns for full value, which is good because Uncle Vernon spent all of his money on the initial set of presents. Given the value of all the presents in the shop, and all of the presents that Uncle Vernon bought, you are to compute the maximum number of presents that could be obtained for Dudley by exchanging presents at the store.

Consider the following small example: Vernon bought Dudley two presents, with values £12 and £14. The store has items that cost £8, £5, and £7. By returning the £14 item and buying the £5 and £8 items, Uncle Vernon is able to increase the number of gifts from 2 to 3 (and he even has £1 left over — shhh... don't tell Dudley). With a little thought, you can probably convince yourself that it's impossible to perform exchanges to have more than 3 presents, so this is the best we can do.

The Problem to Solve

You are given two lists of values: first is a list of the values of all the presents that Uncle Vernon bought, and second is a list of the value of each item in the store. Since Uncle Vernon had to increase the number of gifts each year, these can be very large lists: Uncle Vernon could have initially bought as many as 1000 gifts, and the store could have as many as 10,000 items. Each value will be an integer in the range 1 to 100,000. The values may be in any order, and there may be duplicates (so there can be multiple items with the same value). Your program should be able to process any such input in under 5 seconds.

Hints and Techniques

Think clearly about the problem, and reflect on how greedy Dudley is. As you think about algorithms for this, try to convince yourself that your algorithm will work in all cases.

Input and Output

The input consists of an integer `n`, which is the number of presents initially purchases, followed by `n` lines that give the value of each present. This is followed by an integer `m`, which is the number of items in the store, followed by `m` lines giving the value of each item. Your program should output a line with a single integer: the maximum number of presents that Uncle Dursley can get for Dudley. The sample input below describes the sample problem from the "Background" section.

Sample Input

```
2
12
14
3
8
5
7
```

Sample Output

```
3
```

Problem 5.R2: Making Widgets

Required Problem
Points: 50 points

Background

Imagine a production line, where widgets are produced in parallel by manufacturing machines. Each machine has a maximum rate at which it can produce widgets, but to make the rest of the production line work smoothly we need each machine to be producing widgets at the same rate. This means that the rate of production is controlled by the slowest machine in the production group, and faster machines must slow down to that rate. For example, consider three machines for which the maximum production rate (widgets per minute) of the three machines is 8, 3, and 5. Using the rate 8 machine and the rate 5 machine, we could produce 10 widgets per minute (since both would have to operate at the same speed of 5 widgets/minute). However, adding the third machine does not help: adding a machine with maximum production rate of 3 widgets per minute slows down the other machines, and so the total production rate would only be 9 widgets per minute (3 machines each producing 3 per minute). It would be nice to be able to optimize production in such a situation.

The Problem to Solve

Given a large set of potential machines, you are to compute the maximum total production rate possible using some subset of those machines. All production rates are integers in the range 1 to 1,000,000, and there will be at most 100,000 machines. Your program should be able to process any such input in under 5 seconds.

Input and Output

The input will consist of an integer `n` followed by `n` lines, each containing an integer that represents the maximum production rate of a machine. You should output a line containing a single integer: the maximum possible production rate. The input below represents the problem described in the "Background" section.

Sample Input

```
3
8
3
5
```

Sample Output

```
10
```

Problem 5.C1: Maintaining Connections

Challenge Problem
Ninja Points: This challenge problem is worth up to 20 base ninja points
5 extra ninja points for the fastest solution

Background

Consider maintaining a set of machines that are all working together on a task. For each machine there is a "base cost" for maintaining that machine, and then there is a "per connection" charge for each connection to another machine. All machines are connected to each other, so as machines are added to the set you not only have to pay the maintenance fee for that machine, but the maintainance charge for each existing machine increases (since each must add a connection). The problem is to figure out how many machines you can support with a fixed maintenance budget.

For example, if every machine had a base maintenance cost of $50 and a per-connection maintenance cost of $20, then supporting a single machine would cost $50, supporting two machine would cost $140 (two base charges of $50 each, and one connection for each machine), and supporting three machines would cost $270. If we had a budget of $300 for maintenance, then we could only support 3 machines.

When different machines have different maintenance cost structures, it is harder to compute the maximum number of machines that you can support. Consider the following four machines, where the maintenance budget is $160.

Machine	Base	Per-Conn
Machine 1	10	100
Machine 2	500	1
Machine 3	30	20
Machine 4	40	30

In this case you could support machines 1 and 3, or machines 3 and 4, but there is no way to support more than two machines within your budget.

The Problem to Solve

You are to read in information on the maintenance costs of a set of machines, and output the maximum number of machines that you can support. Base and per-connection maintenance costs are integers, and in the range 0 to 1000, and the total maintenance budget is a positive integer that is at most 100,000,000. There can be as many as 1,000,000 machines, and you must answer in 30 seconds or less.

Hints and Techniques

This problem shares a theme with the two required problems in this problem set, but for a really fast solution you must combine that with the "theme" from one of the previous problem sets!

Input and Output

The input consists of an integer `n`, followed by `n` lines that contain the base and per-connection maintenance costs of each of `n` machines. Following those `n` lines will be a single line that contains the total maintenance budget. Your program should output a single line containing the maximum number of machines that you can support. The sample shown below reflects the second example from the "Background" section.

Sample Input

```
4
10 100
500 1
30 20
40 30
160
```

Sample Output

```
2
```

Problem 5.C2: Card Shark

Challenge Problem
Ninja Points: This challenge problem is worth up to 20 base ninja points

Background

Eida has a photographic memory, and she is playing a game that uses a deck of n different cards containing information that she hasn't seen before. In each round of this game the cards are shuffled (perfectly, so that it's a random permutation) and during the course of the round some number $k \leq n$ cards are revealed (the same number in each round). Given a integers $c \geq 1$ and $r \geq 1$, your goal is compute the probability that Eida has seen exactly c different cards after r rounds.

For example, if $n = 4$ and $k = 1$, then after one round Eida will always have seen exactly one card, so if $r = 1$ and $c = 1$ then the answer is 1. If we consider two rounds, then the probability that Eida sees the same card in the second round that she saw in the first is $\frac{1}{4}$, so if $r = 2$ and $c = 1$ the answer is 0.25 — the only other possibility is that Eida has seen two different cards, so if $r = 2$ and $c = 2$ the answer is 0.75.

The Problem to Solve

Given the four parameters to this problem, n, k, r, and c, you need to compute the probability that Eida has seen exactly c cards after playing r rounds in which each round consists of drawing k cards from a deck with n cards. You are guaranteed that these values will satisfy the following properties:

- $n \leq 20$
- $1 \leq k \leq n$
- $1 \leq c \leq n$
- $1 \leq r \leq 100$

Your program should output the probability (in the range 0..1), with 4 digits of precision after the decimal point. Your program should run in less than 30 seconds.

Hints and Techniques

Simulation will not give you the needed accuracy for this problem. You will need to compute probabilities explicitly (and use double, not floats!).

Input and Output

The input will consist of an integer `t`, followed by `t` lines each containing four values: the n, k, r, and c from the problem description above, in that order. For each set of values, output a line containing the requested probability, always including one digit before the decimal point and 4 digits after the decimal point. The sample below illustrates the three cases described in the "Background" section, plus one other case.

Sample Input

```
4
4 1 1 1
4 1 2 1
4 1 2 2
4 1 2 3
```

Sample Output

```
1.0000
0.2500
0.7500
0.0000
```