Special Homework

CS 624, 2022 Fall

Grading: Each of these problems is worth up to 25 points toward your Midterm Exam 2 score. If adding points to your exam score would raise it above 100, then the added points over 100 are halved. (For example, if you already had a 90 score for the exam and submitted two perfect answers, your effective midterm exam score would be 120/100.)

Review the course homework policies before you start!

Problem 1: How many books can a bag hold before it bursts?

You are testing a new bookbag design for a manufacturer. The Engineering department has provided you with their Academic Environment Simulator for testing, a number of identical prototype bags, and N identical books. They said "When we tested it with N books, it failed. We didn't try any other number. We don't actually even know if it holds 1 book!" The Marketing department wants a precise number.

Let the "book tolerance" be the greatest integer number of books (≥ 0) for which the design succeeds. The initial range of possible tolerance values is $\{0, 1, \ldots, N-1\}$, a set of N elements. When you test a bag with some number m of books, either the bag passes or it fails and the bag is completely destroyed. If a bag passes, it is still in perfect condition (that is, bags do not accumulate damage) and can be reused in other tests; a destroyed bag cannot be reused. All bags are identical, and all books are identical. If a test succeeds for m books, it succeeds for any lesser number of books. If a test fails for m books, it also fails for any greater number of books. Testing is expensive, so you want to **minimize** the number of tests that you run to find it, but you must also plan for the **worst case** where it takes many tests to find the tolerance.

Let $T(k, n)$ be the minimum, worst-case number of tests you must run to discover the tolerance, if you have k bags to use and if the range of possible tolerance values has n elements. For example, if n is 3, that could represent a possible tolerance range of $\{0, 1, 2\}$, or a possible tolerance range of $\{22, 23, 24\}$, or many other possibilities—the actual range is not important, only the size of the range. Clearly $T(k, 0) = 0$: if the range is empty, you must have already found the tolerance. And clearly $T(1, n) = n$: if you only have one bag available, you can't afford to lose it during testing unless its destruction tells you the precise tolerance, so you must test the range one by one in ascending order. But if you have $k > 1$ bags available, you can afford to use one up if it significantly reduces the potential tolerance range. And if you have more...

(a) Suppose you have $k > 1$ bags and a range of $n > 0$ possible tolerance values—let's say $\{0, \ldots, (n-1)\}$, to be specific. Now suppose you run a test with m books.

If the test **succeeds**, what changes, and how many tests might you need to run given what you just learned? If the test **fails**, what changes, and how many tests might you need to run given what you just learned? (Hint: The answers form the core of the *optimal substructure property* for this problem.)

(b) Given that you must handle either success or failure, how do you combine the previous answers to determine $T(k, n)$ if you assume that m is the best test to run?

(c) How do you determine which m is the best test to run?

(d) Based on your previous answers, give a recursive equation for $T(k, n)$ where $k > 1$ and $n > 0$.

Problem 2: Seat guests at tables to maximize happiness.

You are helping to plan a dinner event, and the organizer has made you responsible for seating. The event will have N guests, and each guest is labeled 1 through N. There are K tables. Each table can hold any number of guests. Each guest must be seated at one table. The organizer insists that every table must have at least one guest ("No empty tables!"), and that they also insist that the set of labels of the guests at each table forms a contiguous interval. So for example, a table can have guests labeled $\{1, 2, 3, 4, 5\}$ or $\{13, 14\}$ or even $\{21\}$, but not a group like $\{6, 8, 11\}$. The tables are indistinguishable; all that matters is the partitioning of the guests into K groups.

To complicate matters, though, each guest has a preference for how many people they want to sit with. You are given an array P of length N, and $P[i] = m$ means that guest number i prefers to sit at a table with m total people. If that is the case, they will be *happy*; otherwise, they will be unhappy. Let $h(i, g)$ represent whether guest i would be happy at a table with g total people; it is defined as follows:

$$h(i, g) = \mathbf{1}_g(P[i]) = \begin{cases} 1 & \text{if } P[i] = g \\ 0 & \text{otherwise} \end{cases}$$

You probably cannot make everyone happy, but your goal is to maximize the number of happy guests.

Let $H(n, k)$ be the maximum total happiness for an event with guests $\{1, \ldots, n\}$ and k tables, for a fixed preference array P. The base case is $H(0, 0) = 0$. We'll define illegal states to be infinitely unhappy, so $H(0, k) = -\infty$ when $k > 0$ ("no empty tables"), and $H(n, 0) = -\infty$ when $n > 0$ ("every guest must be seated at a table").

(a) Suppose you have $n > 0$ guests and $k > 0$ tables to handle. And suppose you decide to put the last g guests (that is, the guests labeled $n - g + 1$ through n) together at a table.

How happy is that table? (Give a formula that computes the happiness of that table.)

(b) In the situation described in part (a), what is the maximum happiness of the rest of the guests?

(c) Based on your previous answers, give a recursive equation for $H(n, k)$ where $k > 0$ and $n > 0$.