Practice Problem 1. Coat Check

Suppose that $n$ women check their coats at a concert. However, at the end of the night, the attendant has lost the claim checks and doesn't know which coat belongs to whom. All of the women came dressed in black coats that were nearly identical, but of different sizes. The attendant can have a woman try a coat, and find out whether the coat fits (meaning it belongs to that woman), or the coat is too big, or the coat is too small. However, the attendant cannot compare the sizes of two coats directly, or compare the sizes of two women directly. Describe how the attendant can determine which coat belongs each woman in expected $O(n \log n)$ time. Give a brief analysis of the running time of your algorithm.
Practice Problem 2. The Power of Quicksort

To get concrete evidence of the power of randomized quicksort, you want to create discrete probability distributions of how many comparisons are needed to perform a randomized quicksort on \( n \) elements. Specifically, if we let \( X_n \) be a random variable denoting how many comparisons are needed, you want to determine \( P(X_n = x) \) for all integers \( x \geq 0 \).

For example, the distribution for \( X_3 \) is

\[
X_3 = \begin{cases} 
2 & \text{with probability } \frac{1}{3} \\
3 & \text{with probability } \frac{2}{3}
\end{cases}
\]

because there is a \( \frac{1}{3} \) probability we will only need 2 comparisons (if the chosen pivot is the middle element) and a \( \frac{2}{3} \) probability that we will need 3 comparisons (if the chosen pivot is the smallest or largest element).

The goal for this problem is to come up with an efficient algorithm that computes the distribution for \( X_n \).

(a) Given two random variables \( X \) and \( Y \) whose domains are the non-negative integers such that \( 0 \leq X, Y \leq m \) for a known parameter \( m \) and their probability distributions, give an efficient algorithm that computes the probability distribution for \( X + Y \). State the running time of this algorithm in terms of \( m \).

(b) Give a polynomial-time dynamic programming algorithm that computes the distribution for \( X_n \).

*Hint: Start by expressing \( X_n \) in terms of \( X_0, X_1, \ldots, X_{n-1} \).*
Practice Problem 3. The Sensational 6046

The Sensational 6046 is a group of superheroes in charge of fighting off an alien invasion of Earth. Their most powerful ability involves linking their bodies together to form various weapons (including swords, loganberries, and Turing machines). In the heat of battle, they must be able to form these weapons as quickly as possible. Thanks to your expertise in algorithms, you feel that you can help them.

There are \( n \) heroes numbered 1 through \( n \). Each hero starts at some location in 3D space and, to form a particular weapon, they need to move to new points, which are also numbered from 1 through \( n \). However, in order to maintain the structural integrity of the weapon, a hero must arrive at point \( j \) at or before time \( t \cdot j \) for some fixed constant \( t \). You are given an \( n \times n \) matrix \( T \) where \( T_{ij} \) denotes the amount of time it takes for hero \( i \) to move to point \( j \).

The problem can be put in mathematical terms as follows:

Let \( \sigma_1, \ldots, \sigma_n \) be a permutation of \( 1, \ldots, n \) and \( t \) a positive constant. This permutation is said to be \( t \)-valid if \( T_{i \sigma_i} \leq t \cdot \sigma_i \) for \( i = 1, \ldots, n \).

(a) Give an efficient algorithm that, given \( T \) and \( t \), finds a \( t \)-valid permutation if one exists, or reports that one doesn’t exist.

The commander is mainly interested in knowing how quickly a weapon can be formed, i.e., the smallest value \( t_0 \) such that there exists a \( t_0 \)-valid permutation.

(b) We will call \( t_c \) a critical value if there exist values \( i, j \) such that \( T_{ij} = t_c \cdot j \). Show that \( t_0 \) must be a critical value.

(c) Give an efficient algorithm that finds \( t_0 \).