# CS302 - Problem Set 8

1. Suppose you are searching an array $A$ of length $n$ for an element with value $t$. You may assume $A$ has no repeated elements and that $t$ is in $A$. The strategy *sampling without replacement* works as follows: Let $T = \{1, 2, \ldots, n\}$. Choose ("sample") an element $g \in T$ at random, and check if $A[g] = t$. If it is, return $g$. If it is not, remove $g$ from $T$ (i.e. set $T$ to equal $T - \{g\}$). Repeat this process, removing the guessed index from $T$ each time you guess incorrectly, so the set of indices that you are guessing from gets smaller and smaller over time, until you sample an index $g$ such that $A[g] = t$. (Replacement refers to whether the guessed index is placed back ("replaced") into the guessing set $T$ or removed after it is guessed.)

   (a) What is the sample space $S$ for sampling without replacement for an array of size 3, if the item you are looking for is at position 1? (Please list all elements of the sample space, do not describe in words.)

   (b) Let $R : S \to \mathbb{R}$ be the random variable such that, for an element $\sigma$ of the sample space, $R(\sigma)$ is the number of rounds (guesses) that occur in the algorithm when the sequence of random outcomes is $\sigma$. What is the value of $R$ for each element of $S$ that you listed in the previous part?

   (c) In the case of the length 3 array, consider the indicator random variables $X_i : S \to \{0, 1\}$ where

$$X_i(\sigma) = \begin{cases} 1 & \text{if } \sigma \text{ has at least } i \text{ rounds} \\ 0 & \text{if } \sigma \text{ has less than } i \text{ rounds.} \end{cases} \tag{1}$$

   We can write $R$ as a weighted sum of these indicator random variables:

$$R(\sigma) = \sum_{i=1}^{3} X_i(\sigma). \tag{2}$$

   Choose an element $\sigma \in S$ from part (a), calculate $X_i(\sigma)$ for each $i$, and show that indeed, $R(\sigma) = \sum_{i=1}^{3} X_i(\sigma)$ for that particular element $\sigma$.

   (d) Now consider search without replacement with a length-$n$ array. Please describe the sample space as concisely but as descriptively as possible using English. (Your description should be more precise than, e.g. "The set of possible sequences of random outcomes that occur over the course of the algorithm.")

(e) Now consider a length-$n$ array, and, as in the length-3 example, let $R : S \to \mathbb{R}$ be the number of rounds that occur for a given element of the sample space, and let

$$X_i(\sigma) = \begin{cases} 1 & \text{if } \sigma \text{ has at least } i \text{ rounds} \\ 0 & \text{if } \sigma \text{ has less than } i \text{ rounds.} \end{cases} \tag{3}$$

Using linearity of expectation and the special properties of indicator random variables that we've seen in our analysis of QuickSort, determine $\mathbb{E}[R]$ when we have a length-$n$ array. (Help on determining the probability of the events associated with indicator random variables is on the hints.)

(f) Please comment on the average-case vs best-case vs worst-case runtime of search without replacement.

2. You run a plant that produces sheets of aluminum alloy, and then you cut them to size for customers. Your machine produces rectangular sheets of dimension $Q \times R$, and you can cut any sheet into two smaller sheets by making a vertical or horizontal cut at an integer location. So for example, if you have a $2 \times 4$ sized piece, you have the following options that you could produce from a single cut:

- Two $1 \times 4$ pieces (from a horizontal cut at position 1).
- Two $2 \times 2$ pieces (from a vertical cut at position 2).
- A $1 \times 2$ and a $2 \times 3$ piece (form a vertical cut at position 1).

Say you decide to make the cut that produces a $1 \times 2$ and a $2 \times 3$ piece. You could then subsequently cut the $1 \times 2$ piece into two $1 \times 1$ pieces, and the $2 \times 3$ piece into a $2 \times 1$ piece and a $2 \times 2$ piece, and so on. Or you could stop cutting and sell a piece that you've created.

You can also rotate pieces (so a $2 \times 3$ piece is the same as a $3 \times 2$ piece).

Suppose you produce $n$ different sized products, where product $i$ has dimensions $x_i \times y_i$, and you can sell product $i$ for $v_i$ dollars. (For example, perhaps product 1 is a $2 \times 2$ piece that you can sell for \$2, and product 2 is a $3 \times 5$ piece that you can sell for \$3 dollars.) You can sell multiple copies of the $i$th product if you can produce multiple pieces of that size from your original sheet. Assume $Q$, $R$, $x_i$, $y_i$, and $v_i$ for $i \in \{1, 2, \ldots, n\}$ are positive integers. You will design an algorithm that figures out your maximum profit among any possible sequence of horizontal/vertical cuts.

(a) Please provide pseudocode for a dynamic programming algorithm that outputs your maximum profit. (Note: your code doesn't have to output how you should actually divide the piece, just the profit.) The input to your algorithm should be $(Q, R, x, y, v)$ where $Q$ and $R$ are the size of the piece, array $x$ contains the values $x_i$, array $y$ contains the values $y_i$, and array $v$ contains the values $v_i$.

(b) Explain why your algorithm is correct. (You don't need to write a full proof, but you should describe the logic behind the recurrences that you developed to create your algorithm and explain how your algorithm checks the appropriate cases.)

(c) What is the runtime of your algorithm in terms of $Q$, $R$, and $n$?

(d) Explain briefly how you would modify your algorithm to tell you whether you should divide the current sheet, and if so, where you should make the cut.

3. Here are some problem definitions:

- DOUBLE-3-SAT: Given a 3-SAT-type formula involving the variables $x_1, x_2, \ldots, x_n$ and their negations, are there at least two different satisfying assignments? For example, $(x_1 \vee \neg x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge (\neg x_3)$ has two unique valid assignments, $x_1 = 1, x_2 = 1, x_3 = 0$ and $x_1 = 0, x_2 = 1, x_3 = 0$.
- $k$-INDSET: Given an undirected, unweighted graph $G = (V, E)$, is there a set $V' \subseteq V$ such that $|V'| \geq k$, and for all $v, u \in V'$, there is no edge $\{u, v\} \in E$?
- $k$-CLIQUE: Given an undirected, unweighted graph $G = (V, E)$, is there a set $V' \subseteq V$ such that $|V'| \geq k$, and for all $v, u \in V'$, there is an edge $\{u, v\} \in E$?

(a) Prove DOUBLE-3-SAT is in NP.

(b) Prove $k$-INDSET is in NP.

(c) Prove $k$-CLIQUE is in NP.