By: Eric Koehli, Jacob Chmura, Conor Vedova

Question 1. [1 MARK]

For the following parts of this question, we define the random variable X to be the number of iterations before returning `true`. Also, let $n = |I|$, let k equal the number of copies of x in `A[1..n]`, and $r \in \mathbb{N}$ such that $r \geqslant 1$.

a. What is the probability that this algorithm returns `True` in the first iteration of the repeat loop? Justify your answer.

In the first iteration of the loop, i is chosen uniformly at random from set I. Since the size of $|I| = n$, picking any such i is $\frac{1}{n}$, but we are interested in picking an i such that `A[i] == x`. Since we know that there are exactly k copies of x in `A`, then it must be the case that k indices in `A` map to the value of x. Thus, we can conclude that the probability of the algorithm returning true in one iteration is

$$P(X = 1) = \frac{k}{n}$$

b. What is the probability that this algorithm returns `True`? Justify your answer.

Rephrasing this question, we are trying to find the probability that we find an index i such that `A[i] == x` within r iterations. Since each iteration is independent and since we are not removing anything from I, we find the probability that the algorithm returns `true` as

$$\begin{aligned} P(X \leqslant r) &= P(X = 1) + P(X = 2) + \cdots + P(X = r) \\ &= \frac{k}{n} + \frac{k}{n}\left(\frac{n-k}{n}\right)^1 + \frac{k}{n}\left(\frac{n-k}{n}\right)^2 + \cdots + \frac{k}{n}\left(\frac{n-k}{n}\right)^{r-1} \\ &= \frac{k}{n}\left[\sum_{j=0}^{r-1}\left(\frac{n-k}{n}\right)^j\right] \end{aligned}$$

c. After modifying the algorithm, what is the expected number of loop iterations of this algorithm? Justify your answer.

As seen in the earlier part of the question, X is a random geometric variable, with probability $p = k/n$, where the derivation follows since there are k instances in a sample space of size n. We know from statistics that the expected value of a geometric variable is $E(x) = 1/p$. As such we can treat the problem as the "number of trails until the first success", where a success is measured as as selecting i such that `A[i] == x`, and the number of trails is the number of iterations expected before returning `true`. Thus,

$$\begin{aligned} E(X) &= \sum_{j=1}^{\infty} j \cdot P(X = j) \\ &= \frac{n}{k} \end{aligned}$$

Answered by: Eric
Verified by: Jacob, Conor

Question 2. [1 MARK]

We will implement the answer using disjoint set ADT implemented using forest sets with path compression and weighted union. Begin by creating n trees such that the root of the ith tree has integer value i. This tree represents x_i. Begin by looping over the n input conditions. For each equality constraint, $x_i = x_j$ perform Find(i) and Find(j). Next, union the two set representatives attained. After the first loop, loop again over the input sequences. For each inequality cosntraint, $x_i \neq x_j$, perform Find(i) and Find(j). If these two do not equal, return NIL, else, continue looping. If the entire loop is run, then there is a valid assignment for each x value. Loop over i from 1 to n, and assign $x_i =$ Find(i).

The worst case time complexity of our algorithm is $O(n\alpha(3n, m))$, where $\alpha(3n, m)$ is the doubly recursive function described in class for disjoint set runtime. We achieve this, because we implement a forest structure with path compression and weighted union with a sequence of at most m unions and at most 3n finds (n finds per loop). This runtime is clearly less than $O(nm)$ by nature of the slow growth of $\alpha(3n, m)$.

Answered by: Conor
Verified by: Jacob, Eric