Introduction to Algorithms: 6.006
Massachusetts Institute of Technology — March 24, 2021
Instructors: Mauricio Karchmer, Anand Natarajan, Nir Shavit — Recitation 11

Recitation 11

Warm-up excercises

(a) Given array A of n integers, the Python function below appends all integers from set $\{A[x] \mid 0 \leq i \leq x < j \leq n \text{ and } A[x] < k\}$ to the end of dynamic array B.

```
def filter_below(A, k, i, j, B):
    if (j - i) > 1:
        c = (i + j) // 2
        filter_below(A, k, i, c, B)
        filter_below(A, k, c, j, B)
    elif (j - i) == 1 and A[i] < k:
        B.append(A[i])
```

Argue the **worst-case** running time of `filter_below(A, k, 0, len(A), [])` in terms of $n =$ `len(A)`. You may assume that n is a power of two.

(b) Let T be a binary search tree storing n integer keys in which the key k appears $m > 1$ times. Let p be the lowest common ancestor of all nodes in T which contain key k. Prove that p also contains key k.

Exercise: Sortid Casino

Jane Stock is secret agent 006. She is searching for criminal mastermind Dr. Yes who is known to frequent a fancy casino. Help Jane in each of the following scenarios. (In each case for this part, you may give a worst-case or average-case efficiency, but note which one you are giving.) Note that each scenario can be **solved independently**.

(a) A dealer in the casino has a deck of cards that is missing 3 cards. He will help Jane find Dr. Yes if she helps him determine which cards are missing from his deck. A full deck of cards contains kn cards, where each card has a value (an integer $i \in \{1, \ldots, n\}$) and a suit (one of k known English words), and no two cards have both the same value and the same suit. Describe an efficient[1] algorithm to determine the value and suit of each of the 3 cards missing from the deck.

(b) After determining the locations of the p players with the most chips, Jane observes the game play of each of them. She watches each player play exactly $h < p$ game rounds. In any game round, a player will either win or lose chips. A player's **win ratio** is one plus the number of wins divided by one plus the number of losses during the h observed hands. Given the number of observed wins and losses from each of the p players, describe an efficient algorithm to sort the players by win ratio.

[1] By "efficient", we mean that faster correct algorithms will receive more points than slower ones.

Exercise: Range Pair

Given array $A = [a_0, a_1, \ldots, a_{n-1}]$ containing n **distinct** integers, and a pair of integers (b_1, b_2) with $b_1 \leq b_2$, a **range pair** is a pair of indices (i, j) with $i \neq j$ such that the sum $a_i + a_j$ is within range, i.e., $b_1 \leq a_i + a_j \leq b_2$. Note that parts (a) and (b) can be **solved independently**.

(a) Assuming $b_2 - b_1 < 6006$, describe an $O(n)$-time algorithm to return a range pair of A with respect to range (b_1, b_2) if one exists. State whether your algorithm's running time is average, worst-case, and/or amortized.

(b) Assuming $\max A - \min A < n^{6006}$ (with no restriction on b_1 or b_2), describe an $O(n)$-time algorithm to return a range pair of A with respect to range (b_1, b_2) if one exists. State whether your algorithm's running time is average, worst-case, and/or amortized.

Exercise: Left Smaller Count

Given array $A = [a_0, a_1, \ldots, a_{n-1}]$ containing n **distinct** integers, the **left smaller count array** of A is an array $S = [s_0, s_1, \ldots, s_{n-1}]$ where s_i is the number of integers in A to the left of index i with value less than a_i, specifically:

$$s_i = |\{j \mid 0 \leq j < i \text{ and } a_j < a_i\}|.$$

For example, the left smaller count array of $A = [10, 5, 12, 1, 11]$ is $S = [0, 0, 2, 0, 3]$. Describe an $O(n \log n)$-time algorithm to compute the left smaller count array of an array of n distinct integers. State whether your algorithm's running time is worst-case, amortized, and/or average-case.