Sabancı University
Faculty of Engineering and Natural Sciences

CS301 – Algorithms

Kerat Özyer

Homework 3

00030973

Due: March 19, 2024 @ 23.55
(upload to SUCourse)

S. Ö

PLEASE NOTE:

- Provide only the requested information and nothing more. Unreadable, unintelligible, and irrelevant answers will not be considered.
- Submit only a PDF file. (-20 pts penalty for any other format)
- Not every question of this homework will be graded. We will announce the question(s) that will be graded after the submission.
- You can collaborate with your `TA/INSTRUCTOR ONLY` and discuss the solutions of the problems. However, you have to write down the solutions on your own.
- Plagiarism will not be tolerated.

Late Submission Policy:

- Your homework grade will be decided by multiplying what you normally get from your answers by a "submission time factor (STF)".
- If you submit on time (i.e. before the deadline), your STF is 1. So, you don't lose anything.
- If you submit late, you will lose 0.01 of your STF for every 5 mins of delay.
- We will not accept any homework later than 500 mins after the deadline.
- SUCourse's timestamp will be used for STF computation.
- If you submit multiple times, the last submission time will be used.

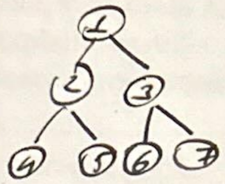
Question 1

Consider a uniquely designed museum where rooms are arranged in a tree structure. Each room can have up to two child rooms connected by a path. Your task is to develop an algorithm to place a minimum number of security guards so that the entire museum is guarded. A guard placed in a room can guard that room, its parent room, and its direct child rooms.

(a) Develop an algorithm to find the minimum number of security guards required for any given museum structured as a standard binary tree. Analyze the worst-case time and space complexity of your algorithm.
Hint: Consider using DFS (for a bottom–up traversal of the rooms).

Answer:

For the purposes of this algorithm, BST node data structure will be slightly modified in order to accomodate for the property that a node will be protected



Pre-order DFS Traversal:
[1, 2, 4, 5, 3, 6, 7]

```
function placeGuards (node)
  left_status = placeGuards(node.left)
  right_status = placeGuards(node.right)
  if left_status := UNGUARDED or right_status := UNGUARDED
    node.guard := GUARDED
    return GUARDED
  if left_status := GUARDED or right_status := GUARDED
    return GUARDED
  else return GUARDED
```

Space complexity is $\Theta(n)$ since an enum variable is introduced. All cases are $\Theta(n)$ since tree must be traversed every time.

(b) Discuss the alterations needed in the algorithm, as well as the changes in worst-case time and space complexity when the museum structure is known to be a red-black tree.

Answer:

In this case, the algorithm may track the red nodes. This is due to the following reason: Red-black trees have the property that red nodes don't have red children and the leaf-root path will always be having the same number of black nodes. This ensures that there will be layers of red nodes that will cover their parents and siblings. The traversal will be still $O(n)$, but no space complexity is introduced.

(c) Given that each room has a number, from the viewpoint of a visitor intending to find/visit room X in the museum starting from the entrance room (i.e., root node), explain the differences experienced when the museum structure is a standard binary search tree versus a red-black tree.

Answer:

Red-black trees are balanced. This means that RB-tree data structure will ensure that the data structure is always at $O(\log n)$ complexity in terms of height. BST's (regular) do not have this property and this causes them to sometimes get linearized. Therefore, lookup operations in BST's will be $\Omega(\log n)$ and $O(n)$, whereas RBT's will be $\Theta(\log n)$. RB structure, finding a specific node will be slightly easier

B. Özen

Question 2

We are given an array A with $2n + 1$ distinct elements. Suppose that we are using the randomized selection algorithm to find the median. In a worst–case scenario of this algorithm, the median is found at the very last step, where each step before that gets rid of only one element in the array. How many different worst–case scenarios are there for finding the median in A?

3rd Smallest

Example: Suppose that we have $A = [4, 5, 1, 3, 2]$. [ ] [2, 3, 4, 5] [ ] [3, 4, 5] [3, 5] [3]

One of the worst–case scenarios is to pick the following elements as the pivots in this order 1, 2, 5, 4. Because (randomly but unluckily) if we pick these numbers in this order as the pivots, we will get rid of only the pivot in each step, and only at the very end (when we have only element 3 remaining, which is the median of the original input array) we will find the median.

The other worst–case scenarios are:

1,5,2,4
1,5,4,2
5,1,4,2
~~4.3.2.1~~ 5,1,2,4
~~2.1.2.1~~ 5,4,1,2
1,2,5,4

5, 4, 1, 2 ⇒ reverses [1 2 3] [5]

[a, a, a, b, b, b]

1 2 3 4 5

$$\frac{4!}{2!\,2!}$$

Answer:

Only one order of selection from the sets of numbers smaller than the median and bigger than the median. The order may be intertwined but internal orders stay the same. I will apply repetitive permutation

$$\Rightarrow \frac{2n!}{n!\,n!}$$

(whole array)

(individual arrays)

Internal sorting of the elements will be discarded with this notation

Question 3

In the WCL Select algorithm, suppose that we modify the approach to partition the array into groups of size 2k+1 instead of the usual groups of 5. Write down the recurrence for the running time of the algorithm for this general case.

Answer:

In case of groups with size 2k+1, we have, $\frac{n}{2k+1}$ total number of subgroups, $\frac{n}{4k+2}$ of which has k+1 elements guaranteed to be smaller than the pivot, which leaves $4k+2-k-1=3k+1$

$\Rightarrow \frac{n(3k+1)}{4k+2}$ elements bigger than the pivot.

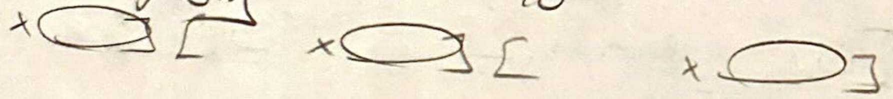
Hence, recursion of this case of WCL-Select is:

$$T(n) = T(n/2k+1) + T\left(\frac{n\cdot(3k+1)}{4k+2}\right) + O(n)$$

$\frac{3n}{10}$ guaranteed bm

$\frac{7n}{10}$

<



$\frac{n}{5}$ groups

3

$\frac{n}{10}$

$\left(\frac{n}{2k+1}\right)$