DALHOUSIE UNIVERSITY

CSCI 6057 - Advanced Data Structures

WINTER 2022

Assignment 3 - SOLUTION

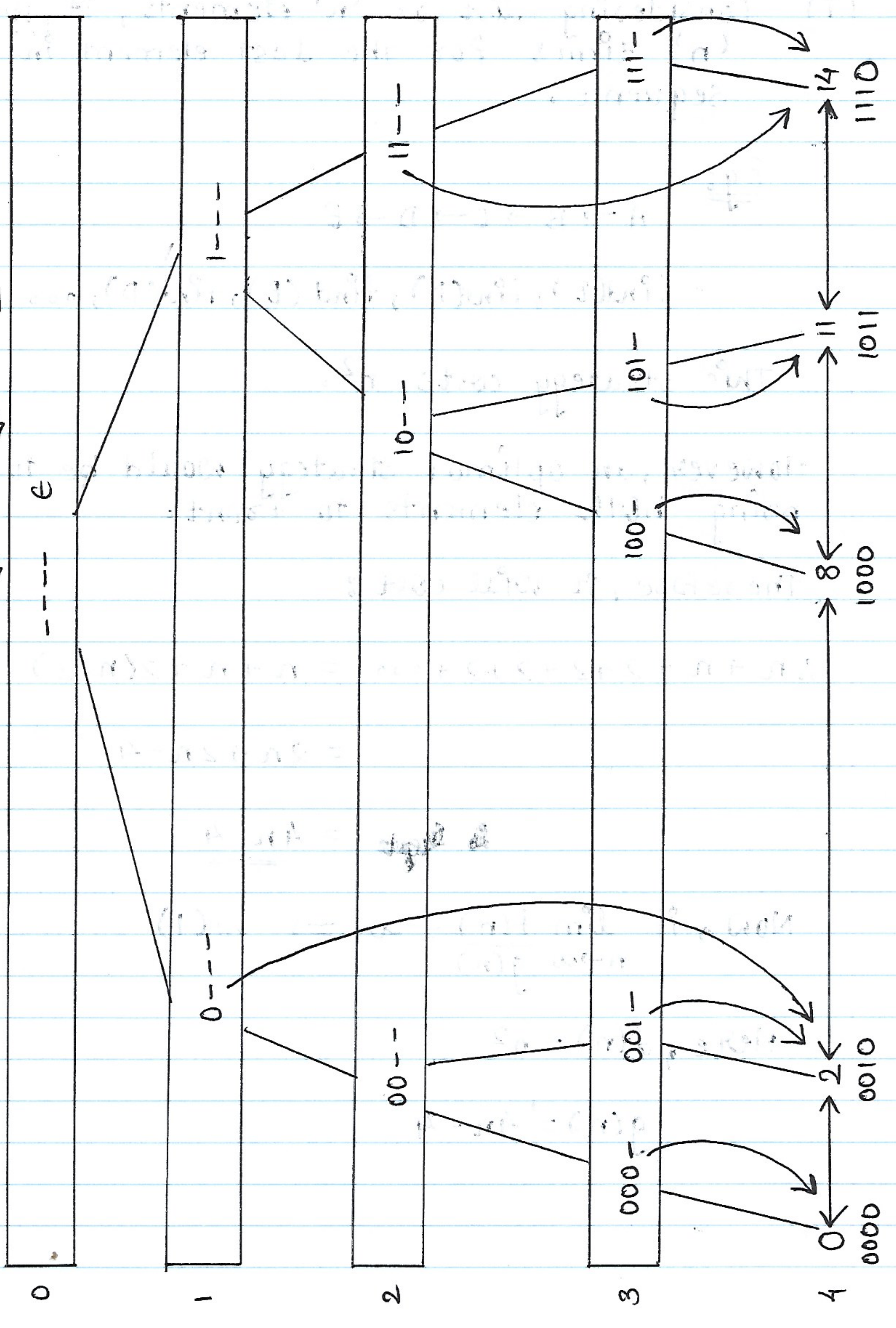
Due: March 2^{nd}, 2022

Banner: B00872269

1

Given: $S = \{0, 2, 8, 11, 14\}$ in universe $\{0, 1, \ldots, 15\}$ $n = 5$ $u = 16$

$\therefore$ hash tables $= \log_2 u = \log_2 16 = \underline{4}$



Keys = [ε, 0, 00, 000, 0000, 0010, 1, 10, 100, 1000, 101, 1011, 11, 111, 1110]

(2)

(2) SOLUTION:

(i) Considering list of 'n' elements, it will ask 'n' times for the last element in the sequence.

Eg:

$$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$$

find(E), find(D), find (E), find(D), ...

This strategy costs n^2.

However, an optimal strategy would be to bring both elements to front.

Therefore, it will cost:

$$\therefore n + n + 2 + 2 + 2 + 2 + \cdots = n + n + 2(n-2)$$

$$= 2n + 2n - 4$$

$$= \underline{4n - 4}$$

Now, if $\lim\limits_{n \to \infty} \frac{f(n)}{g(n)} = \infty \Rightarrow \omega(1)$

Here, $f(n) = n^2$

$$g(n) = \frac{n-1}{2} = S_{opt}$$

③

∴ The ratio of m operations will be :

$$\therefore \lim_{n\to\infty} \frac{n^2}{\frac{4n-4}{2}} \cong \lim_{n\to\infty} \frac{n^2}{4n} = \lim_{n\to\infty} \frac{n}{4} = \infty$$

∴ The ratio of total cost of TR (n^2) and total cost of Sopt ($\frac{4n-4}{2}$) is $\omega(1)$.

(ii) As we saw in (i) that TR always access last item on list :

- For n length of list, for every two accesses it takes 2m access time and costs n^2.

- Now, when we move last two items to the front of the list, the cost is $\cong n$.

Due to such a bad sequence and as stated by Sleator and Trojan (1985) in CACM, T is not competitive.

④

③ <u>SOLUTION:</u>

Given:

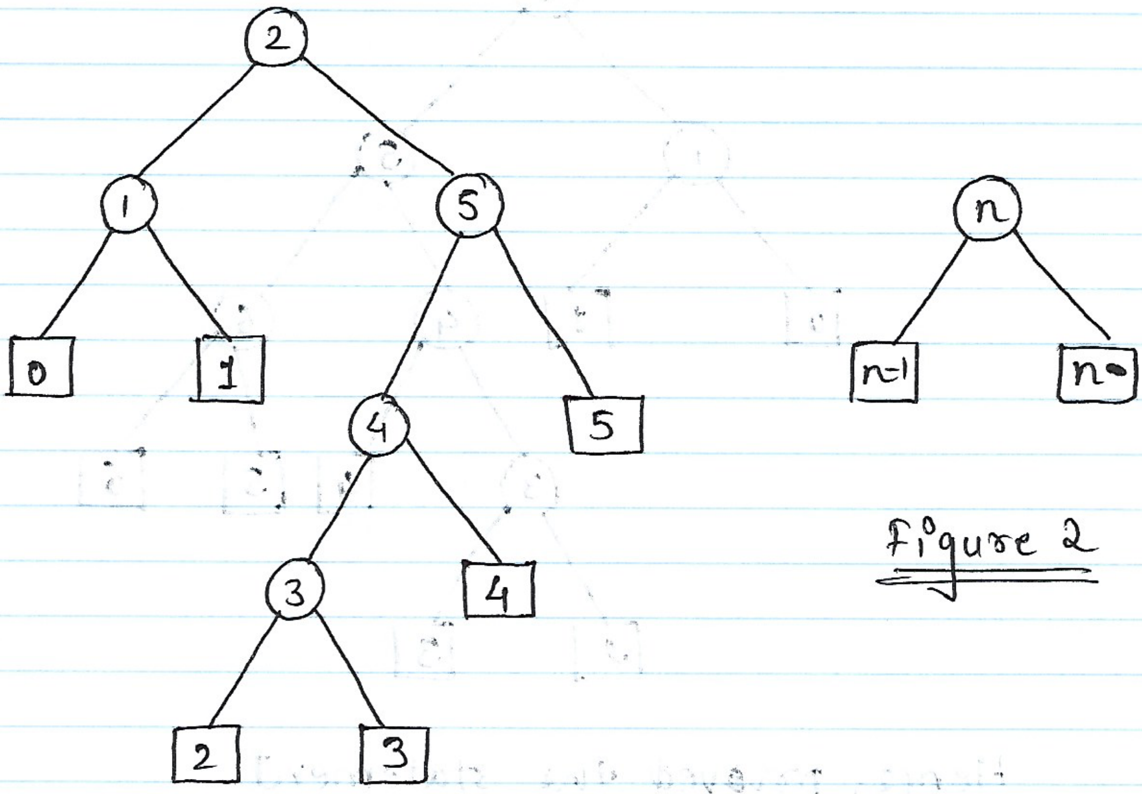
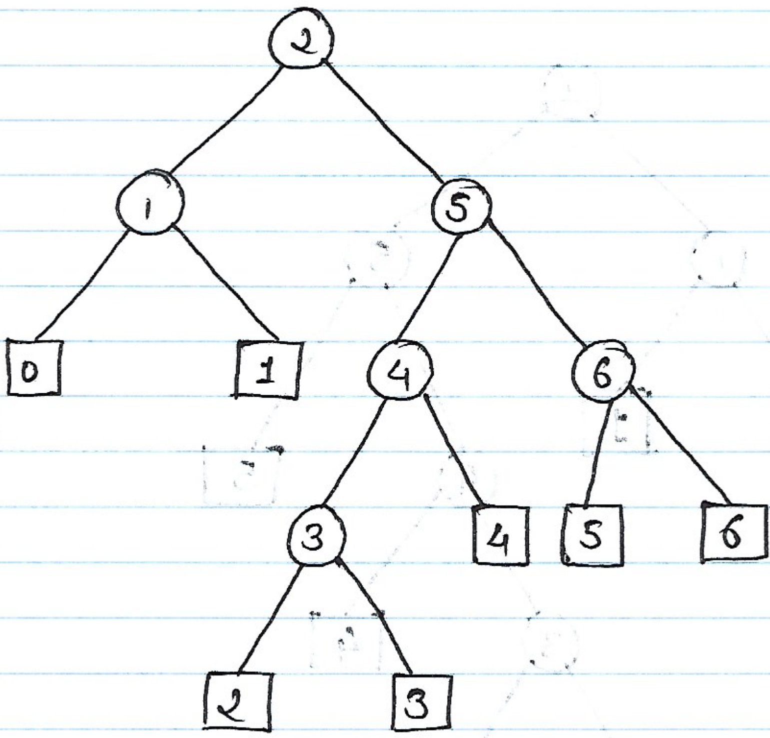


Figure 1

Figure 2

The tree in figure 1 does not contain the very last element because $p_n = q_n = 0$. Thus, for $n-1$ dummy node it will be replaced with subtree from figure 2. The updated tree is on the next page.



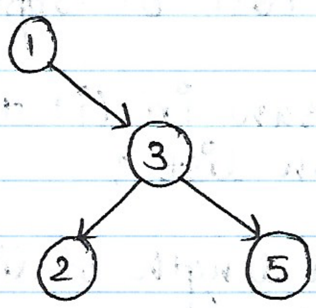
Hence, proved the statement.

6

4 SOLUTION:

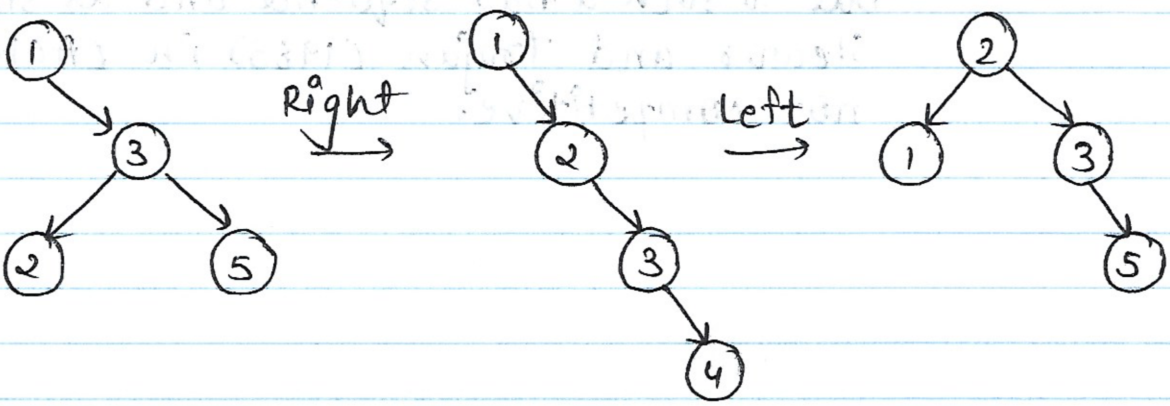
(i) As per the question, all the elements $(A_1, A_2, \ldots, A_n)$ of an arbitary splay tree T are search or accessed one-by-one (i.e. sequentially).

Lets an example as follows:

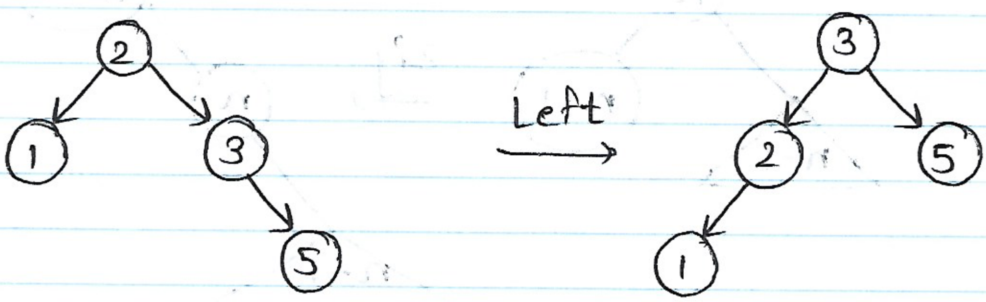


Trying to access 1 ; we get it at ~~[illegible]~~ root. Hence, no need to splay any node.

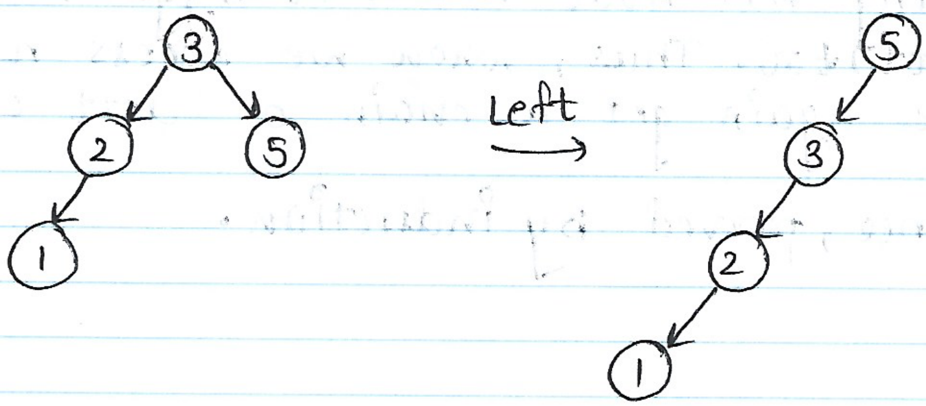
Try to access node 2 :



Try to access node 3 :



Try to access node 5 :

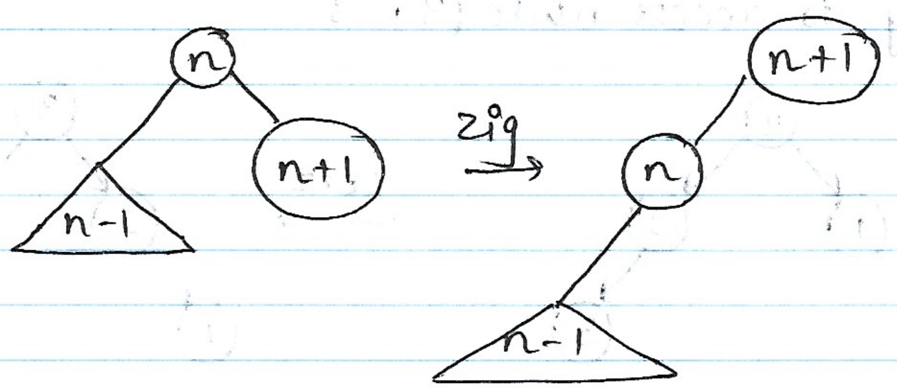


Therefore, the resulting tree only consists of left node chains and thus are $A_1, A_2, \ldots, A_n$ will be placed of T.

(ii) As per (i) the claim holds true for 'n' elements. now lets us proove for n+1.

Since all the nodes 1, 2, ..., n in the splay tree are accessed in sequential order, the resulting tree only consists of left node chain. Therefore, only one position for n+1 node is left : as a right child of the root. See the figure on next page.

⑧



Placing n+1 node makes it bigger than node n (root) too. Thus, when we access n+1 node we again get a chain of left children.

Hence, proved by induction.