

Deeksha Jain
2022CS1099

Date - 20/10/23 classmate

Quiz

CS-203

(Q1) $\Theta(n \log_2 n)$

{ FIND-SUM(S, n)

SORT(S)

i = 0, j = n-1 # n = S.size()

flag = 0

while (i < j)

{

if (S[i] + S[j] == n)

{

~~return TRUE~~; flag = 1, ~~return break~~

else if (S[i] + S[j] < n)

{

i++;

{

else

{

j--;

{

{

if (flag == 1)

return TRUE

{

else

{

return FALSE;

{

$$\underline{Q2} T(n) = 2 T(\sqrt{n}) + \log n$$

$$T(n^{1/2}) = 2 T(\sqrt{\log n}) + \log n^{1/2}$$

$$T(n) = 2 \cdot$$

~~SORT(S)~~

The correctness of the algorithm is proved by proving these condition

- (i) initialization :- before the loop i and j was 0 and $n-1$ which was true initially before the loop.
- (ii) Maintenance :- while The condition $i < j$ valid the loop works
- (iii) Termination :- after $i=j$ loop breaks

SORT func works to take $n \log n$ time

and comparison take 1

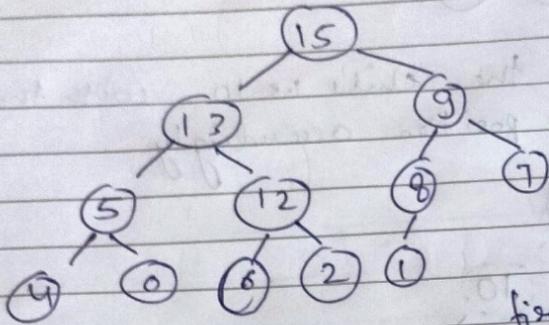
$$T(n) = n + n \log n$$

$$T(n) = \Theta(n \log n)$$

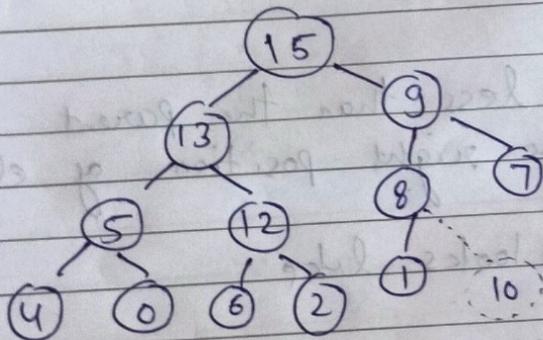
Q3 MAX-HEAP-INSERT ($A, 10$)

$$A = \{15, 13, 9, 5, 12, 8, 7, 4, 6, 10\}$$

The max heap representation of this array is



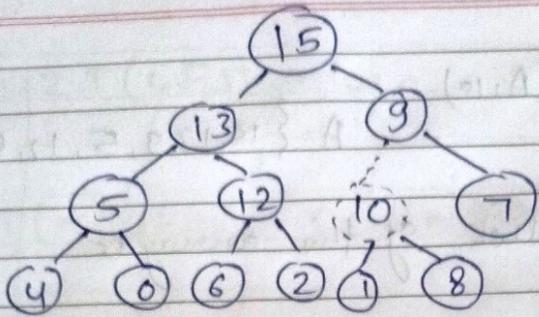
Now for inserting we ~~use~~ insert the element first at last then take it to its right position



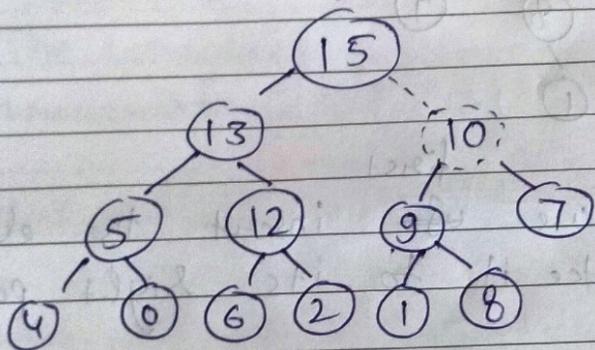
Now compare the child i.e 10 with the parent
if parent > child

No issue

if parent < child
swap(parent, child)



Again compare the child i.e 10 with the parent and change position accordingly



Now child is less than the parent so we've discarded the right position of element

And the array A looks like

$$A = [15, 13, 10, 5, 12, 9, 7, 4, 0, 6, 2, 1, 8]$$

Now here is the algorithm for the same
 MAX-HEAP-INSERT(A, v)

Q2

$$T(n) = 2T(\lfloor \frac{n}{4} \rfloor) + \log n$$

$$T(\sqrt{n}) = 2T(\lfloor n^{1/4} \rfloor) + \log n^{1/2}$$

$$T(n) = 2^k T(\lfloor n^{1/2^k} \rfloor) + k \log n$$

$$\text{let } n = 2^m$$

$$T(\lfloor 2^{m/2^k} \rfloor) = T(2)$$

$$\frac{m}{2^k} = 1$$

$$k = \log m$$

$$k = \log \log n$$