ANALYSIS OF HEAPSORT ALGORITHM

HEAP CREATION WITH INSERT: T(n) = O(n log n)

WITH INSERT
T(n) = O(n log n)

We have first of all the heap creation process which is NlogN time complexity using insert..

First we will consider the number of nodes in a binary tree at a level l---

Level of the tree	Number of nodes
1	1
2	2
3	4

4	8
---	--
I	2^{i-1}

Let us consider the height h of a binary tree of n nodes---

Number of nodes	Height	Estimate
1	1	$Log_2(1+1)$
2	2	$Ceil(Log_2(2+1))$
3	2	$Log_2(3+1)$
4	3	$Ceil(Log_2(4+1))$
5	3	$Ceil(Log_2(4+1))$
6	3	$Ceil(Log_2(4+1))$
7	3	$Ceil(Log_2(4+1))$
8	4	$Ceil(Log_2(5+1))$
9	4	$Ceil(Log_2(5+1))$
----------	---------	
N		$Ceil(Log_2(n+1))$

A node inserted at level l has to move up l-1 levels at most.

2^{l-1} nodes at level l have to move up l-1 levels at most.

For n nodes the number of levels is Ceil(Log_2(n+1)).

T(n) = for level from 1 to Ceil(Log_2(n+1)) do

Move $2^{levels-1}$ nodes levels-1 distance

$$= \Sigma_{1<=levels<=maxlevels}(levels-1)\ 2^{levels-1}, \text{ for maxlevels} = Ceil(Log_2(n+1))$$

Now consider the geometric progression]

$$1 + r + r^2 + r^3 + \text{-----} + r^n = (r^{n+1} - 1)/(r - 1)$$

Differentiating both sides

$$1 + 2r + 3r^2 + 4r^3 + \text{-----} + nr^{n-1} = d/dr[(r^{n+1} - 1)/(r - 1)]$$

Multiplying both sides by r

$$r + 2r^2 + 3r^3 + \text{-----} + nr^n = r\ d/dr[(r^{n+1} - 1)/(r - 1)]$$

$$= [r/(r-1)][n+1]\ r^n - [[(r^{n+1} - 1)/(r-1)^2]$$

$T(N) = [r + 2r^2 + 3r^3 + \text{-----} + nr^n]_{r=2,n=\text{number of levels}}$

$= 2[n+1]\,2^n - [[(2^{n+1} - 1)]$

$= n2^{n+1} + 1$

$< 2\,n2^{n+1} + 1$

$= O(n2^n)$

n = number of levels

$= \text{Ceil}(\log_2(n+1))$

So T(N) = O (N logN) is the time for heap creation.

HEAP CREATION WITH HEAPIFY: T(n) = O(n)

If there are n nodes in the heap then the height of the tree is levels =

$\text{Ceil}(\log_2(n+1))$

.

At a level there are $2^{\text{level}-1}$ nodes, and a node at a level, has to migrate levels-level times.

Hence the time complexity of heapify is

T(n) = [1 node at level 1 has to migrate levels-1 times]

+ [2 nodes at level 2 have to migrate levels-2 times]

+ [4 nodes at level 3 have to migrate levels-3 times]

+---

+ [2^{i-1} nodes at level I have to migrate levels-i times]

$= \Sigma_{1<=i<=levels} 2^{i-1}(levels-i)$

(Let j=levels-i then i= j+levels and i-1=levels-j-1)

$= \Sigma_{1<=j<=levels-1} j2^{levels-j-1}$

$= 2^{levels--1} \Sigma_{1<=j<=levels-1} j2^{-j}$

$= 2^{levels--1} \Sigma_{1<=j<=\infty} j2^{-j}$

[Consider the infinite geometric progression

$1/(1-x) = 1 + x + x^2 + x^3 + x^4 +$ ----

Differentiating both sides we have

$1/(1-x)^2 = 1 + 2x + 3x^2 + 4x^3 + 5x^4 +$ ----

Multiplying both sides by

$x/(1-x)^2 = x + 2x^2 + 3x^3 + 4x^4 + ----$

Putting x =1/2 we have

$2 = \Sigma_{1<=j<=\infty} j2^{-j}$]

So T(n) < 2n = O(n)

HEAPSORT: T(n) = O(n log n)

HEAPSORT
T(n) = O(n log n)

Inserting an element in a heap takes O(logN) time

Deleting an element from a heap takes O(logN) time

In heapsort we delete N elements one after another so the time taken is O(NlogN).

I