Thursday, 15 October 2020 8:23 PM

8 Problem n ropes

→ Join all the ropes to form a single rope.

→ At a time, two ropes can be combined.

cost → $l_1 + l_2$

$A_1 + A_2 \rightarrow 4 + 3 = 7$

Minimum the cost

$6 + 7 = 13 \rightarrow 20$

$5 + 7 + 13$
$= 25$

① Let's say

$A_1 \quad A_2 \quad A_3 \quad A_4$

$A_1 < A_2 < A_3$

→ Find the smallest size ropes

① $A_1 + A_2$

$A_1 + A_2 \quad A_3 \quad A_4$

② $(A_1 + A_2) + A_3$

TC → $(A_1 + A_2) + (A_1 + A_2) + A_3$

②

(a) $A_1 + A_3$

$A_2 + A_4$

(b) $(A_2 + A_4) + A_1$

TC → $(A_2 + A_4) + (A_2 + A_4) + A_1 + A_2$

Find the array

4, 3, 7, 5, 6

sort

2, 3 4, 5, 6

combine → 5

6, 4, 5, 6

sort

4, 5 6, 6

→ 9

9, 6, 6

sort

6, 6 9

→ 12

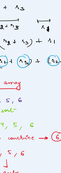
12, 9

sort

→ 9, 12

TC → $O(n \log n)$

Heap



Binary Heap

→ Binary tree

① Complete BST

② All nodes should follow min/max property

5, 10, 7, 18, 13, 1

→ 100 solutions after subtrees

Max Heap

For n nodes, height of CBT → $(\log n)$

Implementations

① Binary tree

② Array

eg Max Heap

root node → arr[1] → max ele

$2i+1$

$2i+2$

$2i+1 = 3$

$i = 2i+1$

$i = 2i+2$

$i = \frac{i-1}{2}$

$i = \frac{i-2}{2}$

$i = \frac{i-1}{2}$ or $\frac{i-2}{2}$

Operations

(Max Heap)

① Insert to Heap

① Insert(2)
② Insert(18) ✓
③ Insert(13)
④ Insert(6) ✓
⑤ Insert(19) ✓
⑥ Insert(8)
⑦ Insert(1)

$2i+1 = 3$

$i = 2i+1$

$i = 2i+2$

$i = \frac{i-1}{2}$

$i = \frac{i-2}{2}$

$i = \frac{i-1}{2}$ or $\frac{i-2}{2}$

6	5	18	19	13	1
18	8	19	5	19	18

TC → $O(\log n)$

$i \rightarrow 2i+1$

$2i+2$

rule → $\left\lfloor \frac{i}{2} \right\rfloor$ parent ele

Root node value eg 100 ↑ ↓ leaves nodes.

② Delete

Max Heap

Movement(1)

10	9	8	5	1

Movement()

5	9	8	1	3

① swap (5, 16)
② delete(5)
③ sink down

TC → $O(\log n)$

Build Heap

① 10 8 5 14 8 30 9 20

add(10) → $(\log n)$
add(8)
add(5)
add(14)
add(8)
add(30)
add(9)
add(20)

TC → $O(n \log n)$

② 5 8 10 14 8 9 30 20

CBT → n nodes

$\frac{n}{2}$ leaf nodes

$\frac{n}{2}$

$\frac{n}{4}$

$\frac{n}{8}$

$\frac{n}{16}$

$\frac{n}{2^h}$

$\frac{n}{2^h} = 1$

$2^h = n$

$h = \log n$

$\frac{n}{2^h} = \frac{n}{2^{\log n}} = \frac{n}{n}$

$= 1$

[45, 16, 25, 65, 12, 48, 18, 1, 100, 28, 4, 2, 95, 9, 30]

For every level → $\frac{n}{2}$ nodes

$S = \frac{n}{2^{h-1}} \times 0 + \frac{n}{2^{h-2}} \times 1 + \frac{n}{2^{h-3}} \times 2 \ldots$

$S = \frac{n}{2^{h-1}}\left(0 + \frac{1}{2} + \frac{2}{4} + \frac{3}{8} + \ldots \frac{h}{2^{h-1}}\right) \rightarrow (1)$

$\frac{S}{2} = \frac{n}{2^{h-1}}\left(\frac{1}{2} + \frac{2}{4} + \frac{3}{8} + \ldots \frac{h}{2^h}\right) \rightarrow (2)$

$(2) - (1)$

$S = \frac{n}{2^{h-1}}\left(1 + \frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} \ldots + \frac{1}{2^{h-1}} - \frac{h}{2^h}\right)$

GP

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

$S = \frac{n}{2^{h-1}}\left(1 \times \frac{\left(1 - \left(\frac{1}{2}\right)^{h-1}\right)}{1 - \left(\frac{1}{2}\right)} + \frac{h}{2^{h-1}}\right)$

$S = \frac{n}{2^{h-1}}\left(\frac{\left(\frac{2^{h-1} - 1}{2^{h-1}}\right)}{1 - \left(\frac{1}{2}\right)} + \frac{h}{2^{h-1}}\right)$

$S = \frac{n}{2^{h-1}} \times \left[\frac{1}{2}\left(\frac{2^{h-1} - 1}{2^{h-1}}\right) \times 2 + h\right]$

$S = \frac{n}{2^{h-1}} \times 2 - 1 + h$

$S = \frac{n}{2^{h-1}} - 2 + h$

$S = \frac{n}{2^{\log n - 1}} - 2 + \log n$

$2^{\log n} = n$

$S = n - 2 + \log n$

$S = O(n)$