Wednesday, 12 April 2023 8:30 PM

Huffman → Heaps

→ Join all heaps to form a single heap.

→ As a binary tree heap can be constructed

cost = $l_1 + l_2$

$n_1 + n_2 \rightarrow n_1 + n_2 = 6$

$6 + 9 + 15 = 30$

"minimize total cost"

$5 + 7 + 9 + 15$

$= 36$

(2) let's say

$n_1 \leq n_2 < n_3$

→ select two smallest size heaps.

(1) $n_1 + n_2$

$n_1 \quad n_2$

(2) $(n_1 + n_2) + n_3$

TC → $\boxed{(n_1 + n_2) + (n_1 + n_2) + n_3}$

→ Sub the wrong

$\boxed{3 \quad 5 \quad 6 \quad 6}$

(1) $\boxed{(3 \quad 5 \quad 6 \quad 6)}$

TC → $O(n^2)$

Heap

Binary heap

→ Binary tree.

(1) complete B.T

(2) ~~the value should greater than parent~~

5, 10, 8, 17, 13, 2

→ no relation b/w two children

$\boxed{2 \quad 5 \quad 8 \quad 6 \quad 17 \quad 13}$

Max heap

For → n nodes

height of ht → $(\log n)$

→ How to build the heap??

(1) Binary tree

(2) Binary search tree (max heap)

Max → at root node

left ≥ 2 ≥ 1

right → 2 ≥ 4

operation

(1) Insert in Heap (min heap)

→ Insert (2)

→ (2)

Temp

$\boxed{2 \quad 5 \quad 8 \quad 6 \quad 17 \quad 13 \quad 20 \quad 25}$

TC → $O(\log n)$

(2) (5) → (2) (pseudo element)

(2) Delete

→ delete

$\boxed{5 \quad 6 \quad 8 \quad 20 \quad 25 \quad 10}$

swap (1, 25)

(25, 10)

(1) swap (1 and 25)

(2) delete (25)

(3) downheapify

TC → $O(\log n)$

→ Build heap

$\boxed{10, 2, 5, 17, 8, 9, 7, 20}$

→ add (10)

→ add (2)

→ add (5)

→ add (17)

→ add (8)

TC → n log n

(1) $\boxed{5, 2, 10, 17, 8, 9, 7, 20}$

145, 109, 25, 65, 12, 48, 18, 2, 109, 27, 4, 8, 15, 9, 20

CBT → n nodes

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

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

2^{h-1} $\quad$ 2^h

For every level → 2^{h-1}

$S = \boxed{2^{h-1} \times 0} + \boxed{2^{h-2} \times 1} + \boxed{2^{h-3} \times 2} + \cdots$

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

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

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

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

GP

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

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

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

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

$S = 2^{\log_2 n} - 2 + h$

$S = n - 2 + \log_2 n$

$\boxed{TC \rightarrow O(n)}$

(1) TC → $O(\log n)$

SC → $O(1)$

$\boxed{2 \quad 5 \quad 8 \quad 6 \quad 17 \quad 13}$

(2) TC → $O(n)$

SC → $O(1)$

$\boxed{2 \quad 5 \quad 8 \quad 6 \quad 17 \quad 13}$

$\boxed{5 \quad 5 \quad 8 \quad 6 \quad 17 \quad 13}$

2 → fixed

complexity | unsorted array | sorted array | sorted linked list | binary heaps

$\frac{n}{2} \rightarrow 4 \quad 10 \quad 3 \quad 2$