Shengjie Shu V00886800
CSC226 Writing-Assn1 Due: Oct.4

1. Lower bounds for sorting.

① seq = $(X_1, X_2 \cdots X_n)$

$\Downarrow$

② seq = $(\underbrace{X_1, X_2 \cdots X_k}_{\text{Subsequence 1}})(\underbrace{X_{k+1}, X_{k+2} \cdots}_{\text{Subsequence 2}}) \cdots (\underbrace{\cdots X_n}_{\text{Subsequence } n/k})$

$\Downarrow$

③ sorting each subsequence:

$(X_1, X_2 \cdots X_k)$ sorted $(X_{k+1} \cdots)$ $(\cdots)$ $\cdots$ $(\cdots X_n)$ sorted.

④ Combine them together

Final sequence = $(X_1, X_2 \cdots X_n)$ sorted.

As supposed, - The input sequence is a concatenation of n/k subsequences each contains k element. (shown above in ②)

- All the elements in each subsequence occur before all the elements in any later subsequence.

(shown above in ③) $\Rightarrow$ which means we only need to consider the sorting algorithm in each subsequence. And don't need to consider the sorting between different sequence.

$\Rightarrow$ Time of sorting each ~~Sub~~subsequence =

$$\Omega(k \log k) \quad \left(\begin{array}{l}\text{each Subsequence} \\ \text{only have } k \text{ elements}\end{array}\right)$$

$\Rightarrow$ Total time of sorting all subsequence =

$$\Omega(n/k \cdot k \log k) = \boxed{\Omega(n \log k)}$$

- After each subsequence is sorted, the concatenation of all the sorted subsequences ~~produced~~ produces a sorted sequence. (shown in ④)

~~$\Rightarrow$ Time = T~~

Thus, the lower bound on the number of comparisons needed to sort this sequence ~~is~~ is $\boxed{\Omega(n \log k)}$

Shengjie Shu V00886800
CSC226 Writing-Assn1 Due: Oct.4

2. Quickselect with median-of-median pivots

~~Use stack of bricks = set $\alpha = \frac{1}{9}$ β~~

~~For~~ each segment of length 9 ; sort to get median . Cost: $O(1)$

$n/9$ such segments : Total cost $O(n)$

Then, there are $n/9$ medians, of which $n/18$ of are less than or equal to m^* (the median of whole list)

For each such median, 4 or more points are less than m^*.

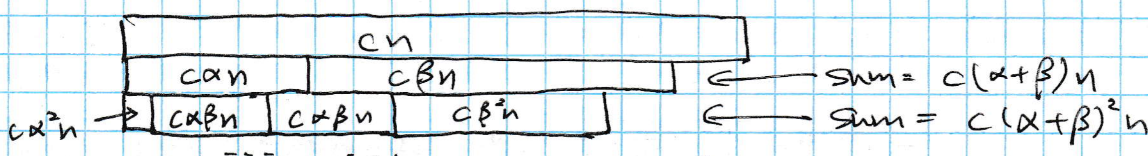
$\Rightarrow$ At least $(5n/18) - 1$ elements less than m^*

Hence, at most $13n/18$ elements are greater than m^*.

By symmetry, at most $13n/18$ elements are less than m^*

$\Rightarrow$ set $\alpha = \frac{1}{9}$ $\beta = 13/18$

Use ~~stat~~ stack of bricks: $T(n) = T(n/5) + T(13n/18) + cn$



$$T(n) \le O(1) + cn \sum_{j=0}^{\infty} (\alpha+\beta)^j$$

$$= \frac{cn}{1-(\alpha+\beta)}$$

$$= \frac{cn}{\frac{3}{18}}$$

$$= 6cn = \boxed{O(n)}$$

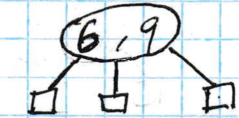
3. Insertion in 2-3 trees.

6, 9. 14. 15. 13. 2. 7. 18. 17

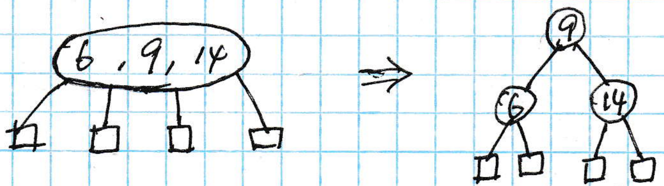
Shengjie Shu V00886800
CSC226 Writing-Assn1 Due: Oct.4

STEP1=
6

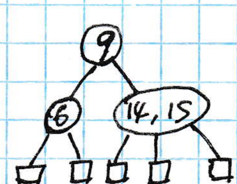
STEP2=
9



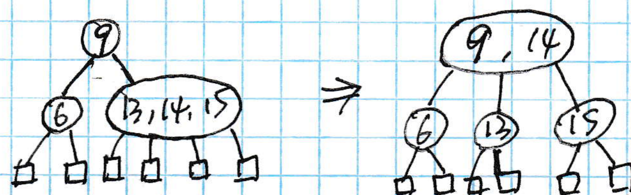
STEP3=
14



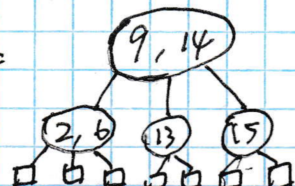
STEP4=
15



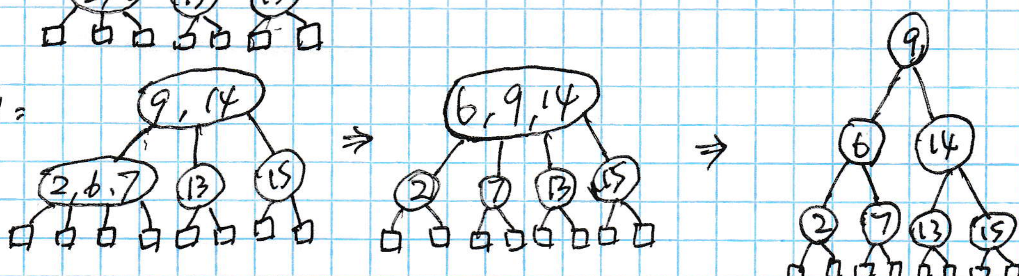
STEP5=
13



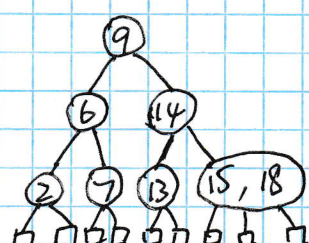
STEP6=
2



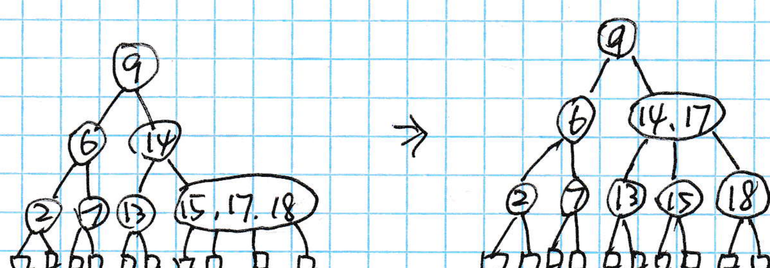
STEP7=
7



STEP8=
18



STEP9=
17



Shengjie Shu V00886800

4. Relaxed AVL trees

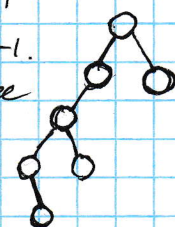
$N(h)$ = the minimum number of elements that can be stored in a relaxed AVL tree.

Prove: $N(h) = \Omega(k^h)$ for some constant $k > 1$.

~~Base case:~~

- Recurrence relation $N(h) = 1 + N(h-1) + N(h-3)$.

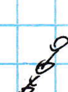
→ As a height h relaxed AVL tree consists of a root node, a left subtree and a right subtree.
→ one of the subtree must be of height $h-1$.
→ the smallest the height of the other subtree can be is $h-3$, since otherwise the height-balance property fails to hold.
→ To minimize the number of nodes the smaller subtree should have height $h-3$, instead of $h-1$.



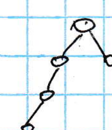
- Base case: $N(1) = 1$

$N(2) = 1 + N(1) = 2 \geq k^2$ Yes

$N(3) = 1 + N(2) = 3 \geq k^3$ Yes



$N(4) = 1 + N(3) + N(1) = 1+3+1 = 5 \geq k^4$



- Induction hypothesis: ~~Suppose for any h, N(h) = Ω(k^h) is true. Then N(h+1) = Ω(k^{h+1})~~

let h_0 be the height of AVL trees, $h_0 > 3$. Assume $N(h_0) = \Omega(k^{h_0})$

- Induction step: ~~we want~~ we want to prove $N(h_0+1) = \Omega(k^{h_0+1})$

$$N(h_0+1) = 1 + N(h_0) + N(h_0-2)$$
$$\geq 1 + k^{h_0} + k^{h_0-2} \quad \text{(by IH)}$$

when $1 + k^{h_0} + k^{h_0-2} \geq k^{h_0+1}$, it's true.

$\Leftrightarrow \quad k^2 + 1 \geq k^3$

$\Leftrightarrow \quad k^3 - k^2 - 1 \leq 0$

$\Rightarrow \quad k \approx 1.4656\ldots$

Thus, when $1 < k \leq 1.4656$, $N(h) = \Omega(k^h)$ for $h > 1$.