Wang Tianduo 1002963 Problem Set 1

Exercise 1 Heap Sort

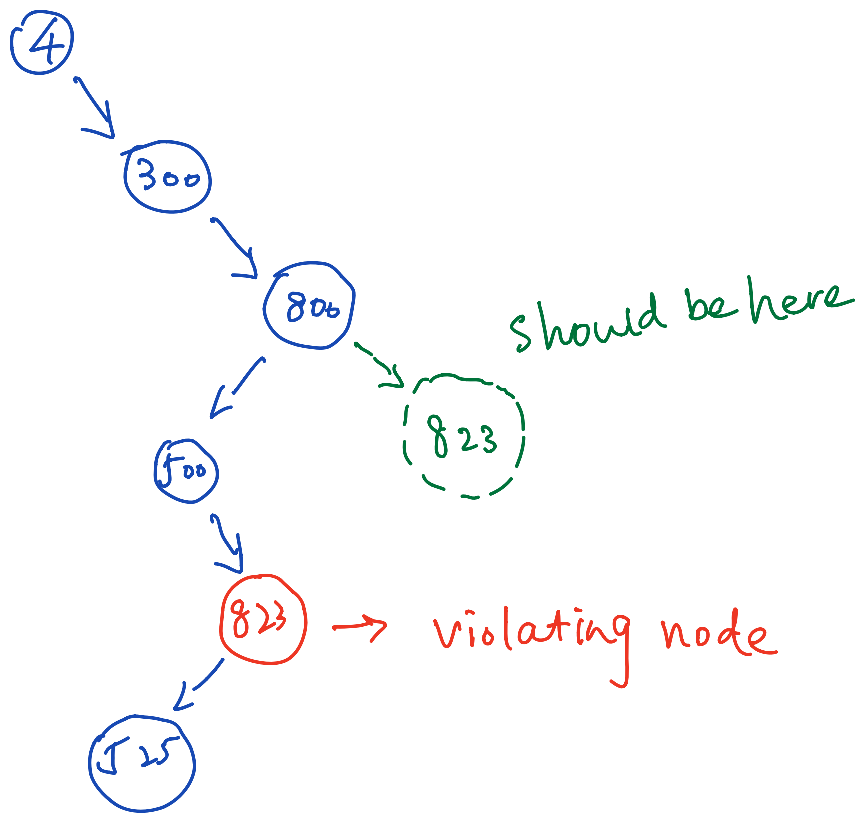
1. Given the height of a heap is h.
   minimum number of elements is 2^h
   maximum number of elements is $2^{h+1}-1$

2. For both sorted array in increasing order and sorted array in decreasing order, running time of HeapSort is $O(n\lg n)$ because new heap needs to be built from array for every iteration.

3. The biggest element can be found in any of the leaves (if the size of heap A is n, then the biggest element could be found in the range $A[\lfloor\frac{n}{2}\rfloor+1 \dots n]$)

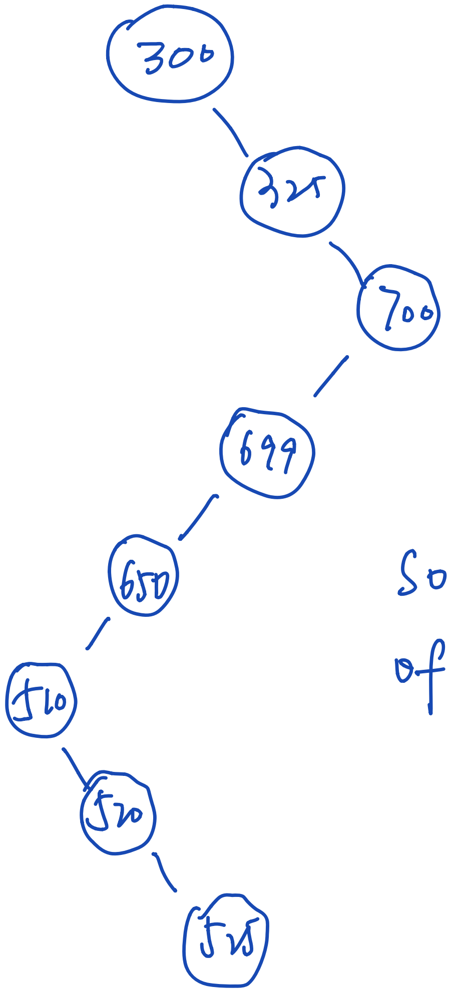
Exercise 2 Binary Search Tree.

(a)



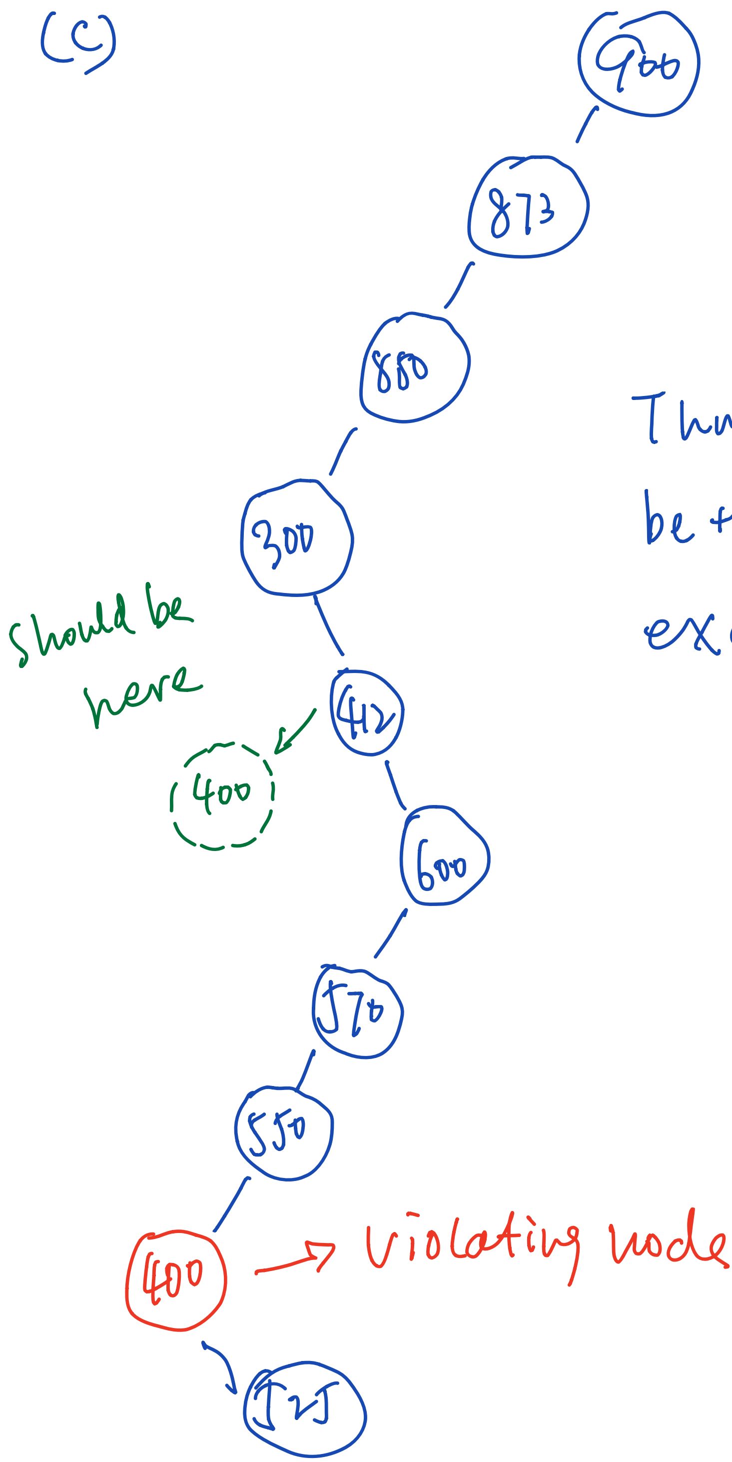
Hence, (a) cannot be a sequence of keys examined

(b)



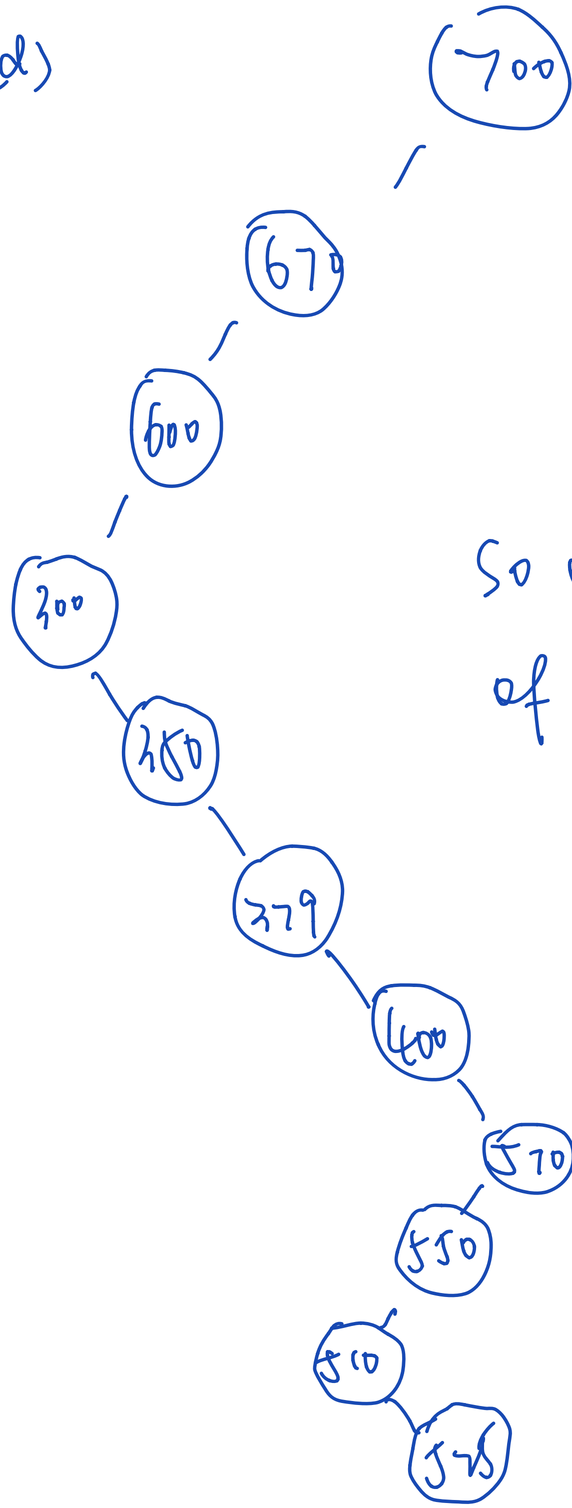
No violating node,
So (b) could be the sequence
of keys examined

(c)



Thus, (c) cannot be the sequence of keys examined

(d)

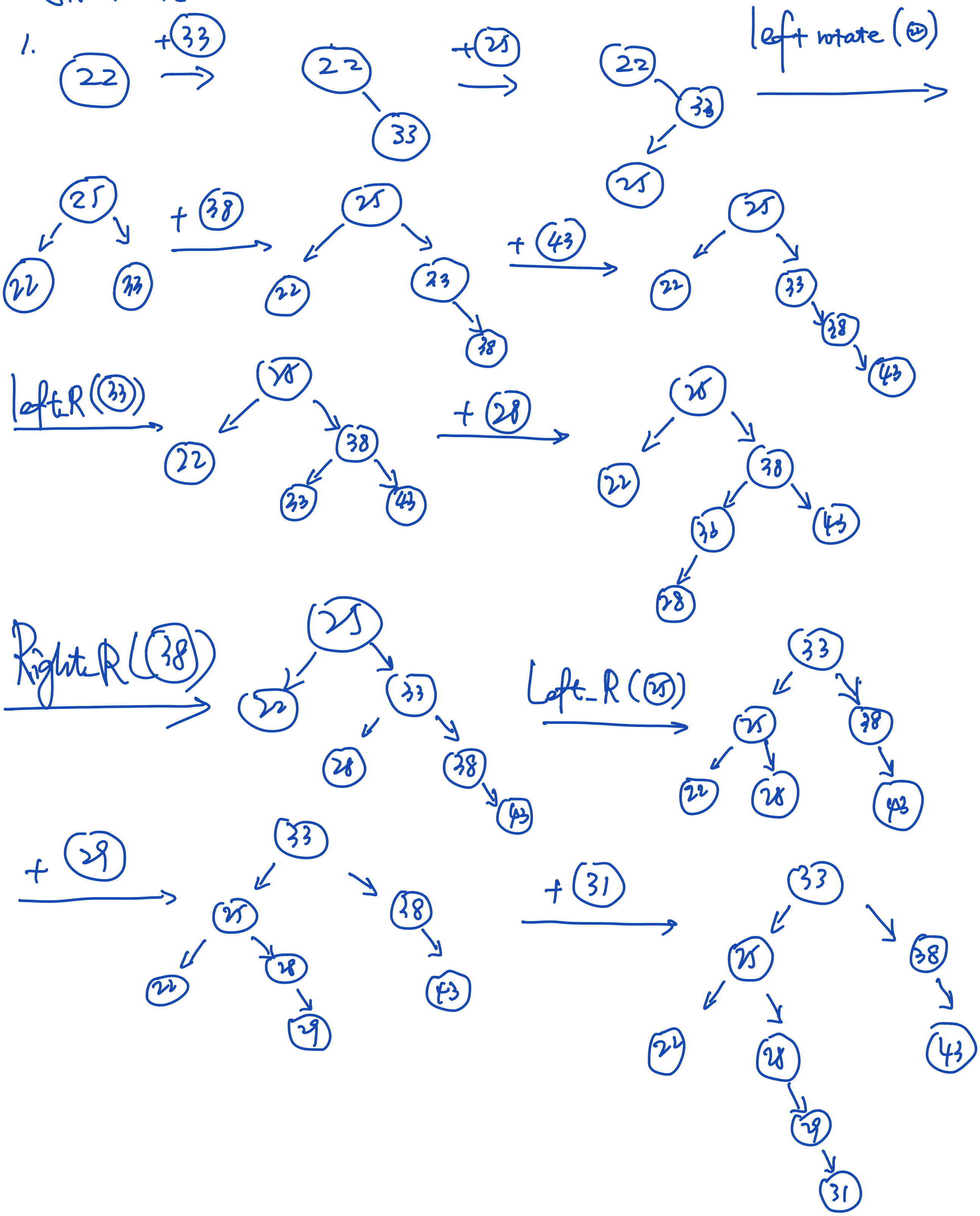


No violating nodes

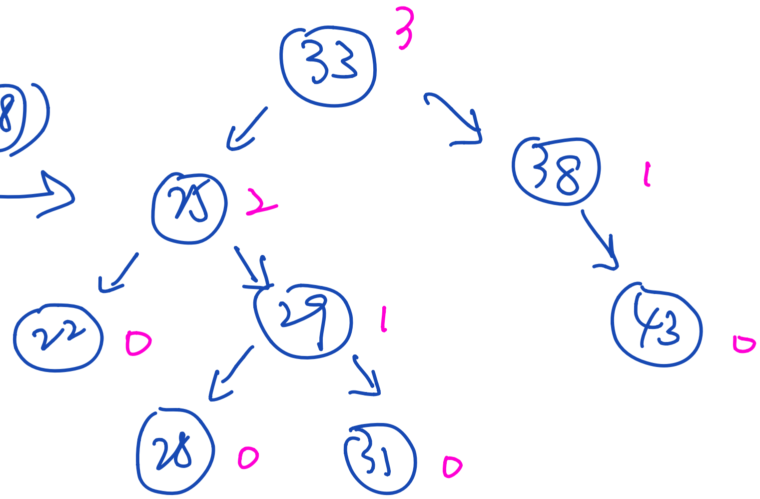
So (d) can be the sequence of keys examined

Exercise 3 AVL Trees.

1.



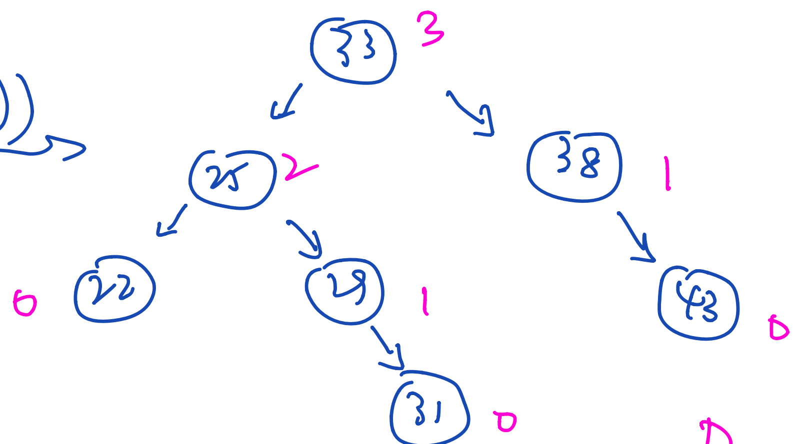
left(28)



Done!

2.

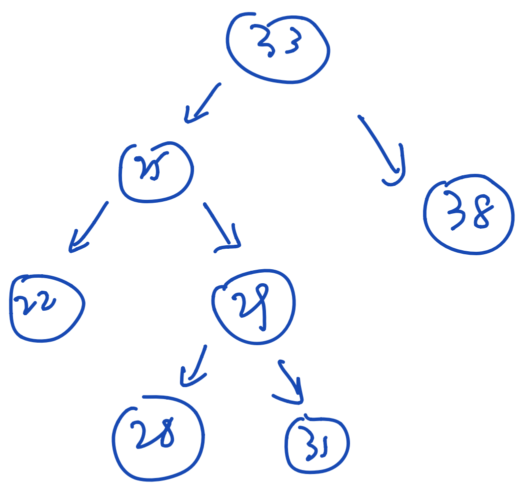
Del(28)



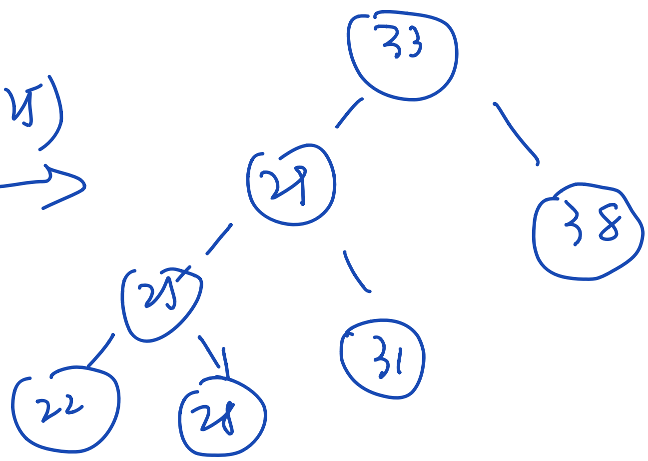
Done!

3.

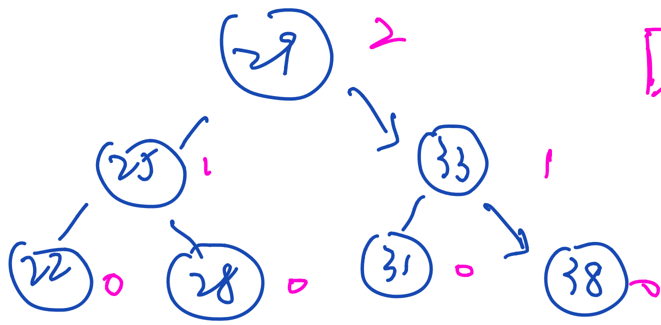
Del(43)



Left_R (25)



Right_R (33)



Done!

Exercise 4 Sorting.

1. Yes, the algorithm will work properly after modification

Reason: The order that elements are taken out of C and put into B doesn't affect the placement of elements with the same key.

2. The algorithm will start by preprocessing exactly as counting-sort so that C[i] contains the number of elements less than or equal to i in the array. When queried about how many integers fall into a range [a...b], simply compute C[b] − C[a−1]. This takes O(1) times and yields desired output.

3.

hat		tea		rag		bat
ten		one		pan		box
hen		rag		hat		hat
two		two		rat		hen
pan		ten		bat		one
one	→	hen	→	tea	→	pan
tea		pan		ten		rag
rat		hat		hen		rat
rag		rat		box		tea
box		bat		one		ten
bat		box		two		two

Exercise 5. Hashing.

Step 1. The initial position probed would be

$$i_0 = T\lfloor h'(k) \rfloor \, , \; j_0 = 0$$

Step 2. $i_1 = (T\lfloor h'(k) \rfloor + j_1) \% m$

$$= (i_0 + 1) \% m$$

$$j_1 = j_0 + 1 = 0 + 1 = 1$$

Step 3.

$$i_2 = (i_2) \% m = (i_1 + j_2) \% m$$

$$= (i_0 + j_1 + j_1 + 1) \% m$$

$$= (i_0 + 1 + j_0 + 2) \% m$$

$$= (i_0 + 1 + 2) \% m \quad (j_0 = 0)$$

Step 4.

$$(i_3)\%m = (i_2 + j_3)\%m = (i_2 + j_2 + 1)\%m$$

$$= (i_1 + j_2 + j_1 + 2)\%m$$

$$= (i_0 + 1 + 2 + 3)\%m$$

$j_0 = 0$, $j_1 = 1$, $j_2 = 2$.

From the above steps, we may conclude

in Step i.

$$h(k, i) = \left(T[h'(k)] + \frac{i(i+1)}{2}\right)\%m$$

$$= \left(T(h'(k)) + \frac{1}{2}i^2 + \frac{1}{2}i\right)\%m$$

$$= (T(h'(k)) + C_1 i + C_2 i^2$$

$\therefore C_1 = \frac{1}{2}$, $C_2 = \frac{1}{2}$