

Function HeapSort (array)

heap = heapify (array) $\leftarrow O(n)$

sorted = new array

n times $\left[\begin{array}{l} \text{for } 1 \dots \text{length} \\ \text{sorted.insert (heap.remove ())} \\ \text{return sorted} \end{array} \right. \log(n)$

end Function

$$O(n \log n + c \cdot n) \\ = O(n \cdot \log n)$$

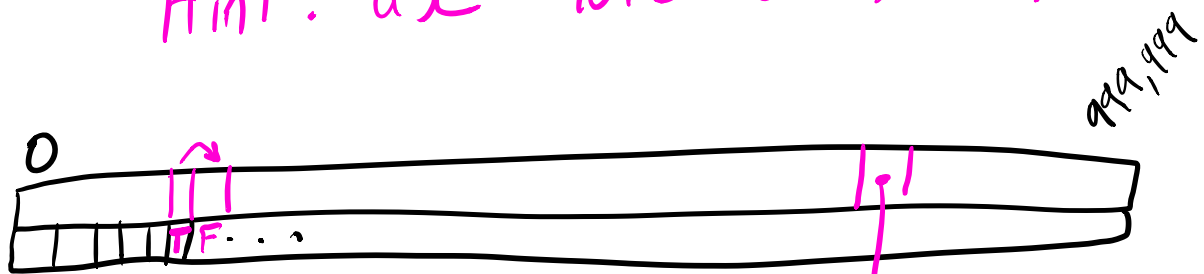
Dictionary Operations

	BST	AVL	hash table
insert	$O(h)$	$O(\log n)$	$O(1) *$
get	$O(h)$	$O(\log n)$	$O(1) *$
remove	$O(h)$	$O(\log n)$	$O(1) *$
update	$O(h)$	$O(\log n)$	$O(1) *$

$$\text{AVL: } h \leq c \cdot \log_2 n \\ h = O(\log n)$$

1. All keys are int
- ~~2. keys $\in [0, \dots, 1000000]$~~
- ~~3. no repeat keys~~

Hint: use lots of memory



insert(5, "a")
get(6)

insert(1000008)

%
range = 8

forward
chaining

linear
probing

Hash function

takes a key, range

gives an int in range