

## 2-11 Heapsort

Hengfeng Wei

hfwei@nju.edu.cn

May 12, 2020



## Obama in a job interview at Google

“What is most efficient way to sort a million 32-bit integers?”

Obama: “The bubblesort would be the wrong way to go.”

Obama: “The bubblesort would be the wrong way to go.”

Best case

Worst case

Average case



$O$

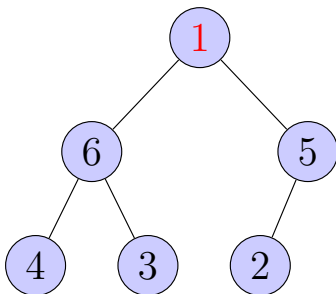
$\Omega$

$\Theta$

## Worst-case of MAX-HEAPIFY (TC 6.2-6)

Show that the **worst-case** running time of MAX-HEAPIFY on an  $n$ -element heap is  $\Omega(\log n)$ .

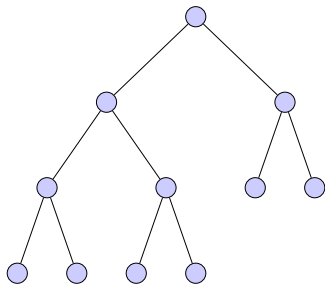
By Example.



COMPARE *vs.* EXCHANGE

## Worst-case of MAX-HEAPIFY (Section 6.2 of CLRS)

Show that the **worst-case** running time of MAX-HEAPIFY on an  $n$ -element heap is  $O(\log n)$ .



$$W(n) \leq H(n)$$

No Examples Here!

Therefore . . .

### Worst-case of MAX-HEAPIFY

Show that the **worst-case** running time of MAX-HEAPIFY on an  $n$ -element heap is  $\Theta(\log n)$ .

	$O$	$\Omega$	$\Theta$
<i>Worst-case</i>	“power” of $\mathcal{A}$	by example	$O = \Omega$

## Worst-case of HEAPSORT (TC 6.4-4)

Show that the **worst-case** running time of HEAPSORT is  $\Omega(n \log n)$ .

By Example.

Non-proof.

$$\underbrace{\Theta(n)}_{\text{EXTRACT-MAX}} \times \underbrace{\Omega(\log n)}_{\text{MAX-HEAPIFY}} = \Omega(n \log n)$$

What is wrong?



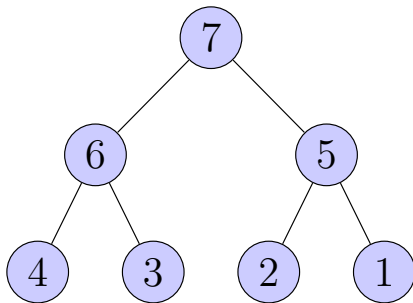
Worst-case of HEAPSORT (TC 6.4 – 4)

Show that the **worst-case** running time of HEAPSORT is  $\Omega(n \log n)$ .

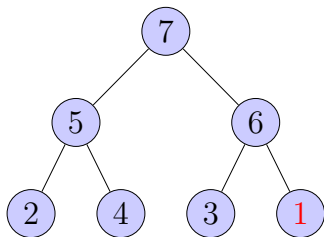




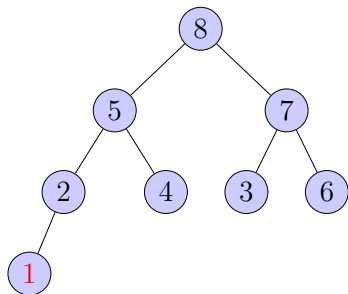
Heap in decreasing order?



$$T(7) = 2 + 1 + 1 + 1 + 0 + 0 = 5$$



$$T(7) = 2 + 2 + 2 + 1 + 1 + 0 = 8$$



(Ex. 23, Section 5.2.3, TAOCP Vol 3)

$$\sum_{r=1}^{n-1} \lfloor \log r \rfloor = n \lfloor \log n \rfloor - 2^{\lfloor \log n \rfloor + 1} + 2 = \Omega(n \log n)$$

## Worst-case of HEAPSORT (TC 6.4 – 4)

Show that the **worst-case** running time of HEAPSORT is  $O(n \log n)$ .

$$\sum_{r=1}^{n-1} \lfloor \log r \rfloor = n \lfloor \log n \rfloor - 2^{\lfloor \log n \rfloor + 1} + 2 = O(n \log n)$$

No Examples Here!

$$\underbrace{\Theta(n)}_{\text{EXTRACT-MAX}} \times \underbrace{O(\log n)}_{\text{MAX-HEAPIFY}} = O(n \log n)$$

Therefore . . .

Worst-case of HEAPSORT (TC 6.4 – 4)

Show that the **worst-case** running time of HEAPSORT is  $\Theta(n \log n)$ .

	$O$	$\Omega$	$\Theta$
<i>Worst-case</i>	“power” of $\mathcal{A}$	by example	$O = \Omega$

## Algorithm $\mathcal{A}$

Inputs  $\mathcal{I}$  of size  $n$

	$O$	$\Omega$	$\Theta$
<i>Best-case</i>	by example	“weakness” of $\mathcal{A}$	$O = \Omega$
<i>Worst-case</i>	“power” of $\mathcal{A}$	by example	$O = \Omega$

### Best-case of HEAPSORT (TC 6.4-5<sup>★</sup>)

Show that when all elements are distinct, the best-case running time of HEAPSORT is  $\Omega(n \log n)$ .

### Best-case of HEAPSORT (Ex. 32, Section 5.2.3, TAOCP Vol 3)

Prove that the number of heapsort promotions,  $B$ , is always at least  $\frac{1}{2}N \log N + O(N)$ , if the keys being sorted are distinct.

## Best-case of HEAPSORT (TC 6.4-5\*)

Show that when all elements are distinct, the best-case running time of HEAPSORT is  $\Omega(n \log n)$ .

Consider the largest  $m = \lceil n/2 \rceil$  elements.

The largest  $m$  elements form a subtree.

$\geq \lfloor m/2 \rfloor$  of  $m$  must be nonleaves of that subtree.

$\geq \lfloor m/2 \rfloor$  of  $m$  appear in the first  $\lfloor n/2 \rfloor$  positions.

They must be promoted to the root before being EXTRACT-MAX.

$$\sum_{k=1}^{\lfloor m/2 \rfloor} \lfloor \log k \rfloor = \frac{1}{2} m \log m + O(m)$$

$$B(n) \geq \frac{1}{4} n \log n + O(n) + B(\lfloor n/2 \rfloor) \implies B(n) \geq \frac{1}{2} n \log n + O(n)$$

## Best-case of HEAPSORT (TC 6.4 – 5)

Show that when all elements are distinct, the **best-case** running time of HEAPSORT is  $O(n \log n)$ .

By Example.





# “On the Best Case of Heapsort” (1994)

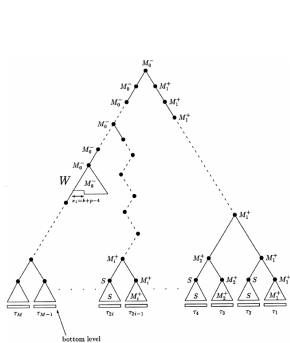


FIG. 2. Initial heap (more detailed).

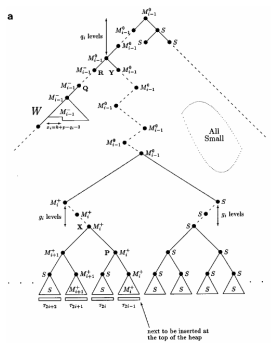


FIG. 3. (a) Odd  $i$ ; (b) contents of the bottom level of  $\tau_{2i-1}$ ,  $i$  odd.

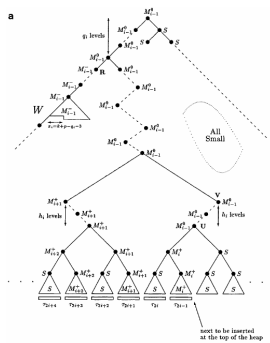


FIG. 4. (a) Even  $i$ ; (b) contents of the bottom level of  $\tau_{2i-1}$ ,  $i$  even.

$$B(n) \leq \frac{1}{2}n \log n + O(n \log \log n)$$

Therefore . . .

### Best-case of HEAPSORT (TC 6.4 – 5)

Show that when all elements are distinct, the **best-case** running time of HEAPSORT is  $\Theta(n \log n)$ .

	$O$	$\Omega$	$\Theta$
<i>Best-case</i>	by example	“weakness” of $\mathcal{A}$	$O = \Omega$

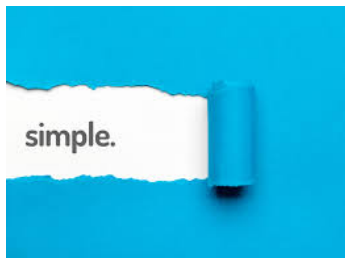
## Algorithm $\mathcal{A}$

Inputs  $\mathcal{I}$  of size  $n$

	$O$	$\Omega$	$\Theta$
<i>Best-case</i>	by example	“weakness” of $\mathcal{A}$	$O = \Omega$
<i>Worst-case</i>	“power” of $\mathcal{A}$	by example	$O = \Omega$
<i>Average-case</i>	$\leq$	$\geq$	$O = \Omega$

## Average-case of HEAPSORT

Assume that all elements are distinct. Show that the **average-case** running time of HEAPSORT is  $\Theta(n \log n)$ .



I said simple,  
not easy.

“By a surprisingly short counting argument.”



Robert Sedgwick



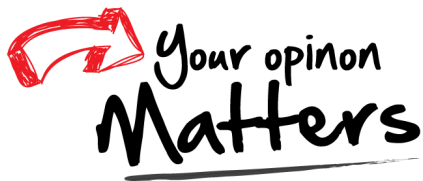
D. E. Knuth

“It is elegant. see exercise 30.”

## Heap Identity

$$\forall h \geq 1 : \lceil \log(\lfloor \frac{1}{2}h \rfloor + 1) \rceil + 1 = \lceil \log(h + 1) \rceil$$

Thank  
You!



Office 302

Mailbox: H016

hfwei@nju.edu.cn