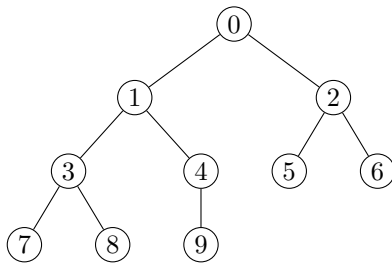


Data Structures Quiz 1

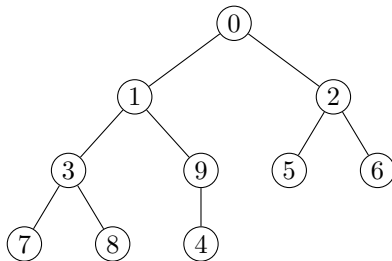
Soundarya Lahari Murari

September 2023

1 Heapify

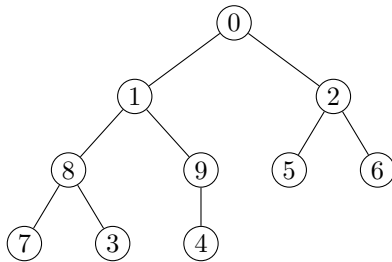


The program begins with $n = 10$, $i = \frac{n}{2} - 1 = 4$. It enters the heapify function.
left = 9, right = 10, largest = 4
left = 9 < 10 and arr[left] = 9 > arr[largest] = 4, so largest = 9
comparison count = 1
right = 10 = 10, so it doesn't enter this part of the if statement
largest = 9 != 4, so 4 and 9 are swapped in the heap.
swapcount = 1
i = 9
left = 19, right = 20, largest = 9
since 19 > 10, 20 > 10, and largest = 9, we break the loop and exit the function.

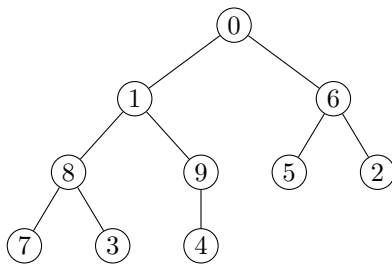


i = 3
left = 7, right = 8, largest = 3
7 < 10 and arr[7] > 3, so largest = 7

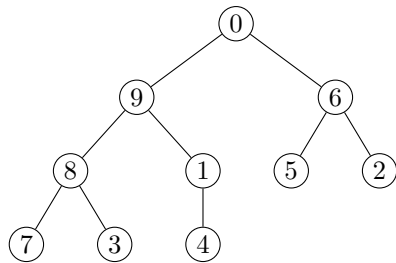
comparison count = 2
 8 < 10 and arr[8] > 8, so largest = 8
 comparison count = 3
 largest != 3, so swap 3 and 8
 swapcount = 2
 i = 8 since 17 > 10, 18 > 10, and largest = 8, we break the loop and exit the function.



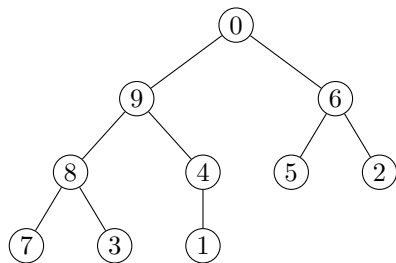
i = 2
 left = 5, right = 6, largest = 2
 5 < 10 and arr[5] > 2, so largest = 5
 comparison count = 4
 6 < 10 and arr[6] > 5, so largest = 6
 comparison count = 5
 largest != 2, so swap 6 and 2
 swapcount = 3
 i = 6 since 13 > 10, 14 > 10, and largest = 6, we break the loop and exit the function.



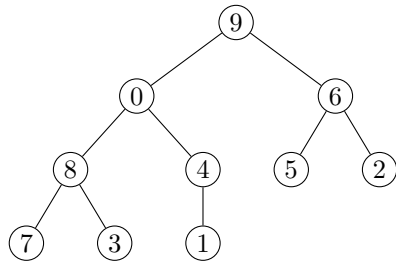
i = 1
 left = 3, right = 4, largest = 1
 3 < 10 and arr[3] = 8 > 1, so largest = 3
 comparison count = 6
 4 < 10 and arr[4] = 9 > 8, so largest = 4
 comparison count = 7
 largest != 1, so swap 1 and 9
 swapcount = 4



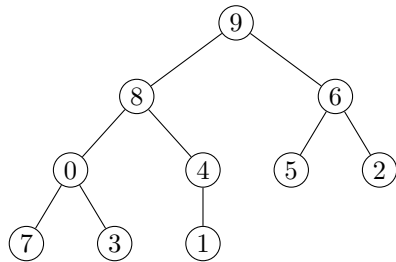
$i = 4$
 $\text{left} = 9, \text{right} = 10, \text{largest} = 4$
 $9 < 10$ and $\text{arr}[9] = 4 > 1$, so $\text{largest} = 9$
 $\text{comparison count} = 8$
 $\text{right} = 10 = 10$, so it doesn't enter this part of the if statement
 $\text{largest} \neq 4$, so swap 1 and 4
 $\text{swapcount} = 5$
 $i = 9$
 since $19 > 10, 20 > 10$, and $\text{largest} = 9$, we break the loop and exit the function.



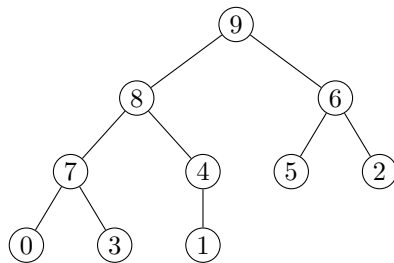
$i = 0$
 $\text{left} = 1, \text{right} = 2, \text{largest} = 0$
 $1 < 10$ and $\text{arr}[1] = 9 > 0$, so $\text{largest} = 1$
 $\text{comparison count} = 9$
 $2 < 10$, but $\text{arr}[2] = 6 < 9$, so largest remains $= 1$
 $\text{comparison count} = 10$
 $\text{largest} \neq 0$, so swap 0 and 9
 $\text{swapcount} = 6$



$i = 1$
 left = 3, right = 4, largest = 1
 $3 < 10$ and $\text{arr}[3] = 8 > 0$, so largest = 3
 comparison count = 11
 $4 < 10$, but $\text{arr}[4] = 4 < 8$, so largest remains = 3
 comparison count = 12
 largest $\neq 1$, so swap 0 and 8
 swapcount = 7
 $i = 3$



$i = 3$
 left = 7, right = 8, largest = 0
 $7 < 10$ and $\text{arr}[7] = 7 > 0$, so largest = 7
 comparison count = 13
 $8 < 10$, but $\text{arr}[8] = 3 < 7$, so largest remains = 7
 comparison count = 14
 largest $\neq 3$, so swap 0 and 7
 swapcount = 8 $i = 7$
 since $15 > 10$, $16 > 10$, and largest = 7, we break the loop and exit the function.



The array has now been sorted into a heap. There have been 8 swaps and 14 comparisons.

This is an example of the worst case. The worst case complexity of the heapify algorithm is $O(n)$.

2 Heap Insert

elm = 0, so we're putting 0 in the heap

arr[0] = 0, i = 0

exits because i = 0



elm = 1

arr[1] = 1



i = 1

parent = 0

arr[parent] = 0 < 1, so 0 and 1 are swapped.

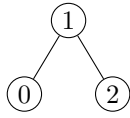
comparison count = 1

swap count = 1

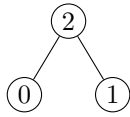


elm = 2

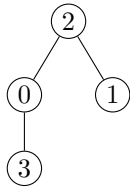
arr[2] = 2



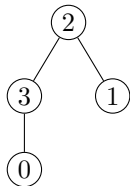
$i = 2$
 $\text{parent} = 0$
 $\text{arr}[\text{parent}] = 1 < 2$, so 1 and 2 are swapped.
 $\text{comparison count} = 2$
 $\text{swap count} = 2$
 $i = 0$, so breaks and exits function



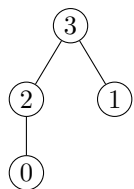
$\text{elm} = 3$
 $\text{arr}[3] = 3$



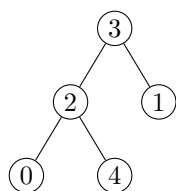
$i = 3$
 $\text{parent} = 1$
 $\text{arr}[\text{parent}] = 0 < 3$, so 0 and 3 are swapped.
 $\text{comparison count} = 3$
 $\text{swap count} = 3$
 $i = 1$



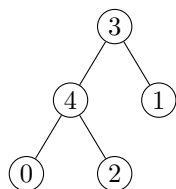
$i = 1$
 $\text{parent} = 0$
 $\text{arr}[\text{parent}] = 2 < 3$, so 2 and 3 are swapped.
 $\text{comparison count} = 4$
 $\text{swap count} = 4$
 $i = 0$, so breaks and exit function



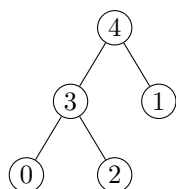
elm = 4
arr[4] = 4



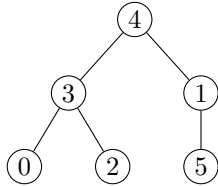
i = 4
parent = 1
arr[1] = 1 < 4, so 2 and 4 are swapped.
comparison count = 5
swap count = 5
i = 1



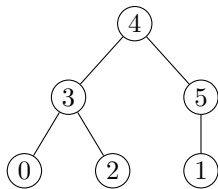
i = 1
parent = 0
arr[0] = 3 < 4, so 3 and 4 are swapped.
comparison count = 6
swap count = 6
i = 0, so breaks and exit function



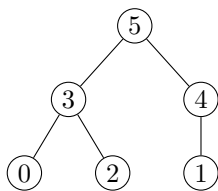
elm = 5
arr[5] = 5



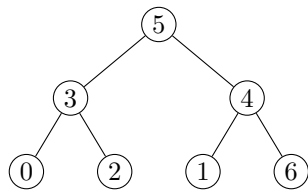
i = 5
parent = 2
arr[2] = 1 < 5, so 1 and 5 are swapped.
comparison count = 7
swap count = 7
i = 2



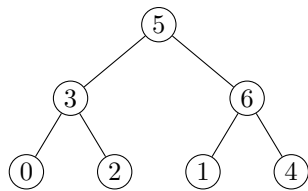
i = 2
parent = 0
arr[0] = 4 < 5, so 4 and 5 are swapped.
comparison count = 8
swap count = 8
i = 0, so breaks and exit function



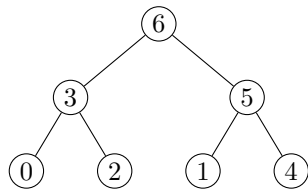
elm = 6
arr[6] = 6



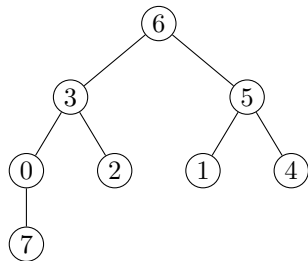
$i = 6$
 $\text{parent} = 2$
 $\text{arr}[2] = 4 < 6$, so 4 and 6 are swapped.
 $\text{comparison count} = 9$
 $\text{swap count} = 9$
 $i = 2$



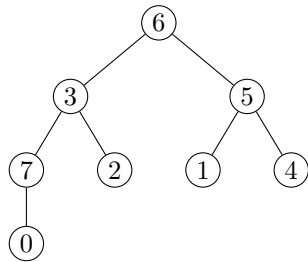
$i = 2$
 $\text{parent} = 0$
 $\text{arr}[0] = 5 < 6$, so 5 and 6 are swapped.
 $\text{comparison count} = 10$
 $\text{swap count} = 10$
 $i = 0$, so breaks and exit function



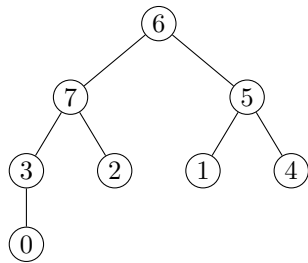
$\text{elm} = 7$
 $\text{arr}[7] = 7$



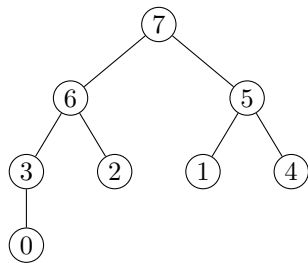
$i = 7$
 $\text{parent} = 3$
 $\text{arr}[3] = 0 < 7$, so 0 and 7 are swapped.
 $\text{comparison count} = 11$
 $\text{swap count} = 11$
 $i = 3$



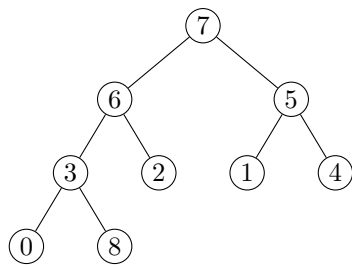
$i = 3$
 $\text{parent} = 1$
 $\text{arr}[1] = 3 < 7$, so 3 and 7 are swapped.
 $\text{comparison count} = 12$
 $\text{swap count} = 12$
 $i = 1$



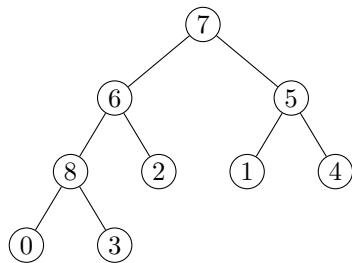
$i = 1$
 $\text{parent} = 0$
 $\text{arr}[0] = 6 < 7$, so 6 and 7 are swapped.
 $\text{comparison count} = 13$
 $\text{swap count} = 13$
 $i = 0$, so breaks and exit function



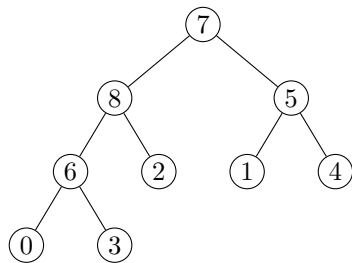
elm = 8
arr[8] = 8



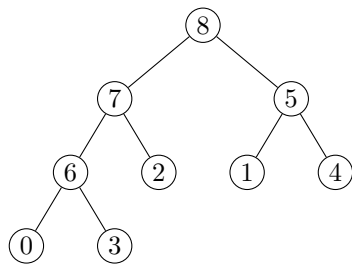
i = 8
parent = 3
arr[3] = 3 < 8, so 3 and 8 are swapped.
comparison count = 14
swap count = 14
i = 3



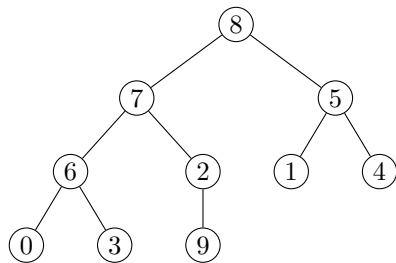
i = 3
parent = 1
arr[1] = 6 < 8, so 6 and 8 are swapped.
comparison count = 15
swap count = 15
i = 1



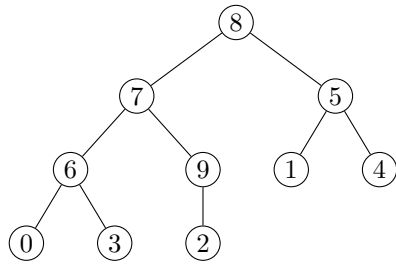
$i = 1$
 $\text{parent} = 0$
 $\text{arr}[0] = 7 < 8$, so 7 and 8 are swapped.
 $\text{comparison count} = 16$
 $\text{swap count} = 16$
 $i = 0$, so breaks and exit function



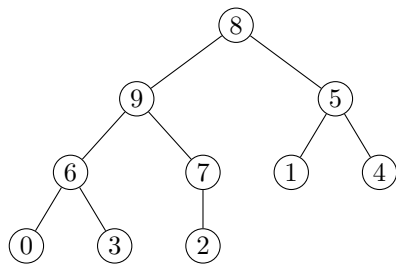
$\text{elm} = 9$
 $\text{arr}[9] = 9$



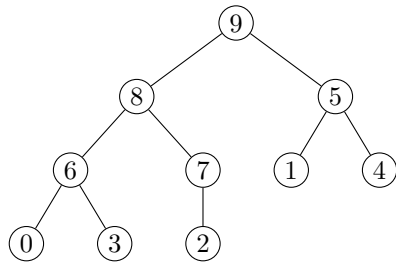
$i = 9$
 $\text{parent} = 4$
 $\text{arr}[4] = 2 < 9$, so 2 and 9 are swapped.
 $\text{comparison count} = 17$
 $\text{swap count} = 17$
 $i = 4$



$i = 4$
 $\text{parent} = 1$
 $\text{arr}[1] = 7 < 9$, so 7 and 9 are swapped.
 $\text{comparison count} = 18$
 $\text{swap count} = 18$
 $i = 1$



$i = 1$
 $\text{parent} = 0$
 $\text{arr}[0] = 8 < 9$, so 8 and 9 are swapped.
 $\text{comparison count} = 19$
 $\text{swap count} = 19$
 $i = 0$, so breaks and exit function



The heap is now finished and the number of comparisons is 19 and the number of swaps is 19 as well. This is an example of the worst case. The worst case time complexity is $O(n \log n)$, because inserting one element is $O(\log n)$ and there are n elements to insert.