

# HEAP CODING DEMO

---

Problem Solving with Computers-II

The image shows the C++ logo in blue, followed by a snippet of C++ code in a monospaced font. The code is: 

```
#include <iostream>
using namespace std;
int main(){
    cout<<"Hola Facebook!n";
    return 0;
}
```

# How is PA 2 going?

- A. Done!    B. On track to finish and having fun.    C. On track to finish but struggling (a bit).  
D. Falling behind and struggling a lot.    E. Haven't read the assignment.

I can deal with pressure, and deadlines.



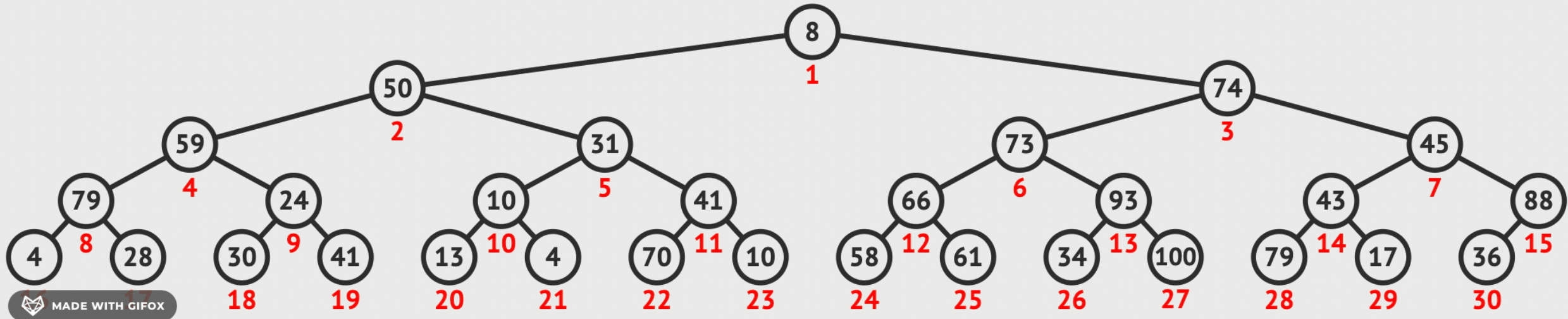


## Helpful links for PA 2

- Find information by reading C++ documentation:
  - **set**: <https://www.cplusplus.com/reference/set/set/set/>
  - **vector**: <https://www.cplusplus.com/reference/vector/vector/?kw=vector>
  - **list**: <https://cplusplus.com/reference/list/list/?kw=list>
  - **stack**: <https://cplusplus.com/reference/stack/stack/?kw=stack>
- Evaluate the time and space complexity of your solutions
  - <https://www.cplusplus.com/reference/set/set/find/>
  - [https://www.cplusplus.com/reference/set/set/lower\\_bound/](https://www.cplusplus.com/reference/set/set/lower_bound/)



# Heapify: A fast way to turn an arbitrary vector to a heap



**High-level approach:** Given an arbitrary vector of keys. Starting from the internal node with the largest index in the vector, and moving upwards in the tree through all the internal nodes (level by level), sift the root of each subtree downward as in the **bubble-down process** until the **heap property** is restored.

**Heapify the vector below to convert it into a max-heap (3 min)**

1	5	3	6	4	1	7	8	4
---	---	---	---	---	---	---	---	---

**What is the resulting vector?**

- A. 8 7 6 5 4 4 3 1 1
- B. 8 1 7 5 4 1 3 6 4
- C. 8 6 7 5 4 1 3 1 4
- D. Something else

# Heap Sort Algorithm

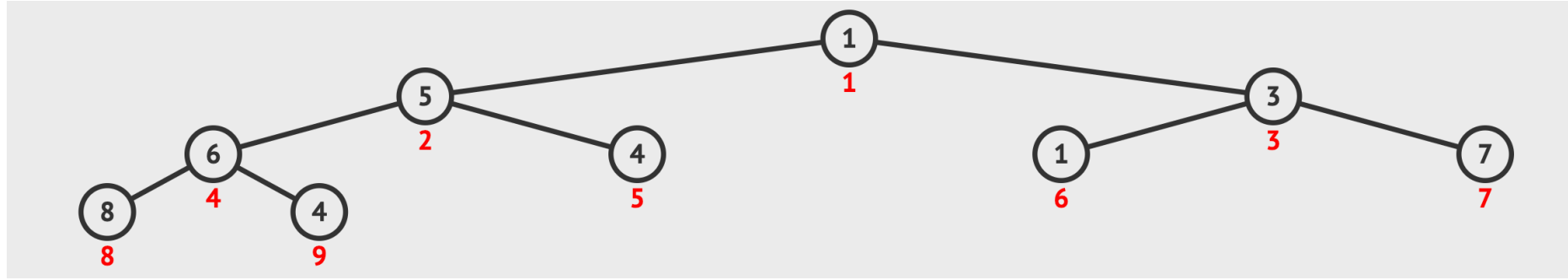
- Step 1: Heapify the input vector with  $n$  keys
- Step 2: Let  $S$  be the number of keys in the heap. Extract the max element (root key) by swapping it with the last key in the vector. Reduce the size of the heap by 1. At this point, the first  $(S - 1)$  keys in the vector represent the heap and the remaining are the sorted portion of the vector. Finally, restore the heap property of the root using the bubble down process
- Repeat step 2 while the size of the heap is greater than 1.

## **Activity 3: Running time of heapify (10 min)**



In a complete binary tree of height  $h$ , how many total nodes are there in levels  $0$  to  $l$

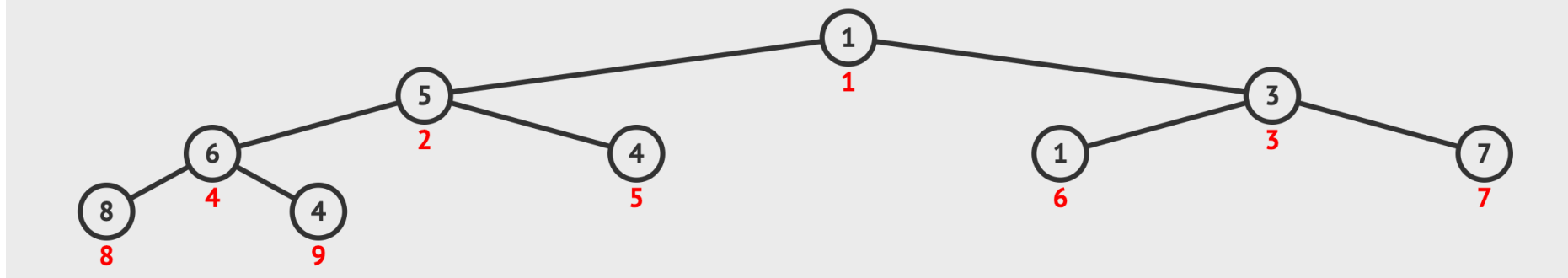
Assume  $l \leq h - 1$



- A.  $2^l$
- B.  $2^l + 1$
- C.  $2^l + 1 - 1$
- D.  $2^h + 1 - 1$
- E. Something else

In a complete binary tree of height  $h$ , what is the index of the last node at level  $l$

Assume  $l \leq h - 1$



A.  $2^l - 1$

B.  $2^{l+1} - 1$

C.  $2^{l+1} - 2$

D.  $2^h + 1 - 2$

E. Something else

In a complete binary tree of height  $h$ , what is the index of the first node at level  $l$

Assume  $l \leq h - 1$

