COMP2054 Tutorial Session 6: Hash Maps, Heaps, and BSTs

Rebecca Tickle
Warren Jackson
AbdulHakim Ibrahim

Session outcomes

- Understand difference between heaps and BSTs
- Know how to apply the various operations on heaps and BSTs
- Understand how hash maps work and deal with collisions

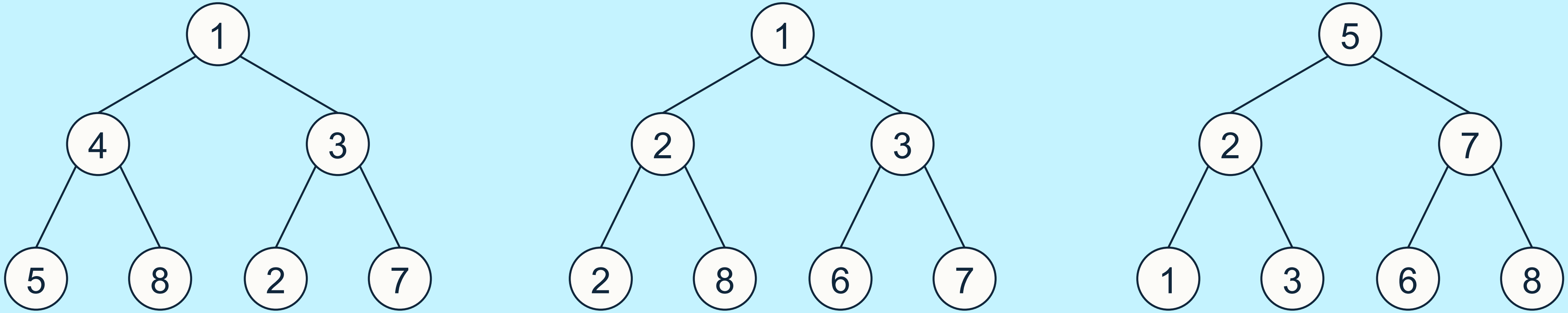
University of Nottingham
UK | CHINA | MALAYSIA

Trees

Heaps and BSTs

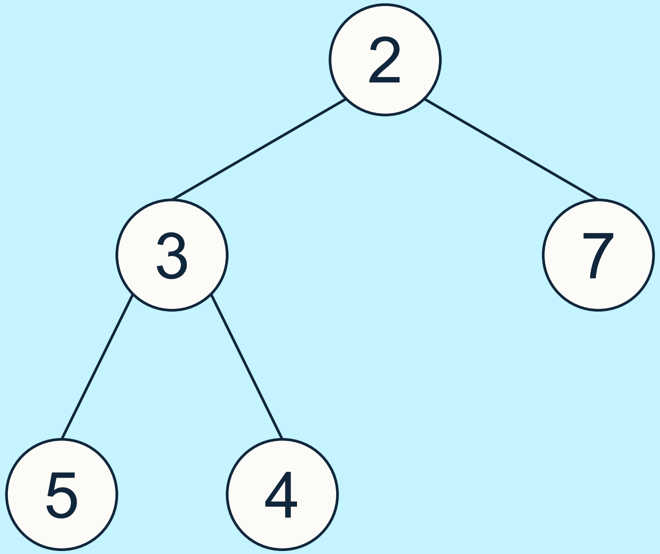
Quiz – Q1

Classify each of the following trees as a Heap, a BST, or neither:



Quiz – Q2

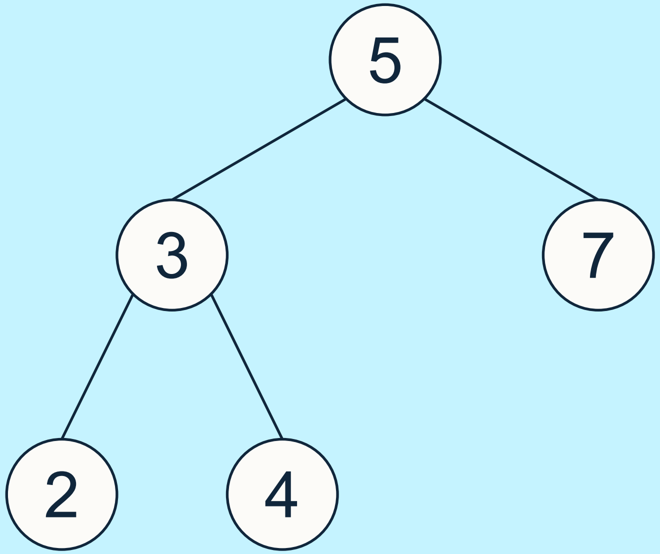
Which two nodes in the following heap should be swapped to create a BST?



Array Representation of Binary Trees

How can we represent the below BST in an array?

Index	[0]	[1]	[2]	[3]	[4]	[5]
Value						



Remember:

```
index(root) = 1
index(left_child, i) = 2i
index(right_child, i) = 2i + 1
```

Quiz – Q3

Given the two binary trees in an array-based format, identify which is a heap and which is a BST, and explain why.

Binary tree 1:

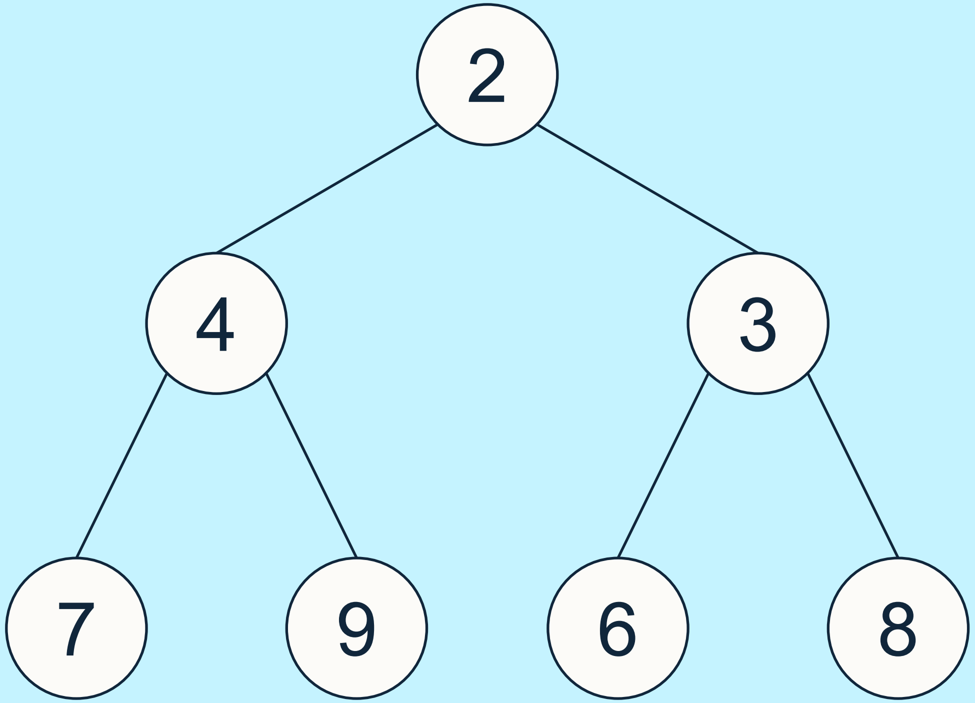
Index	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
Value		2	3	5	4	6		

Binary tree 2:

Index	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
Value		4	3	5	2			6

Operations on Heaps – insertItem(x)

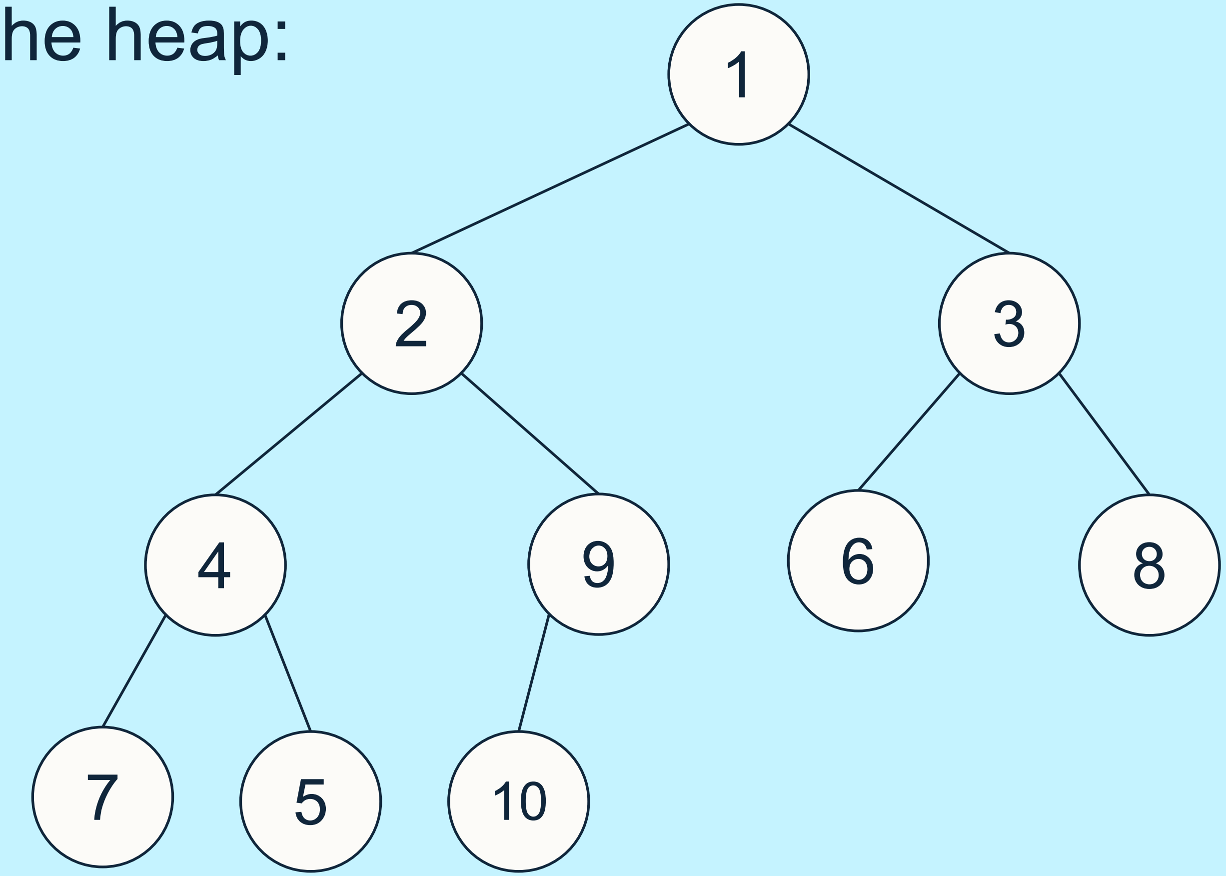
- Starting with the heap:



Perform insertItem(5), insertItem(1), then insertItem(10).

Operations on Heaps – removeMin()

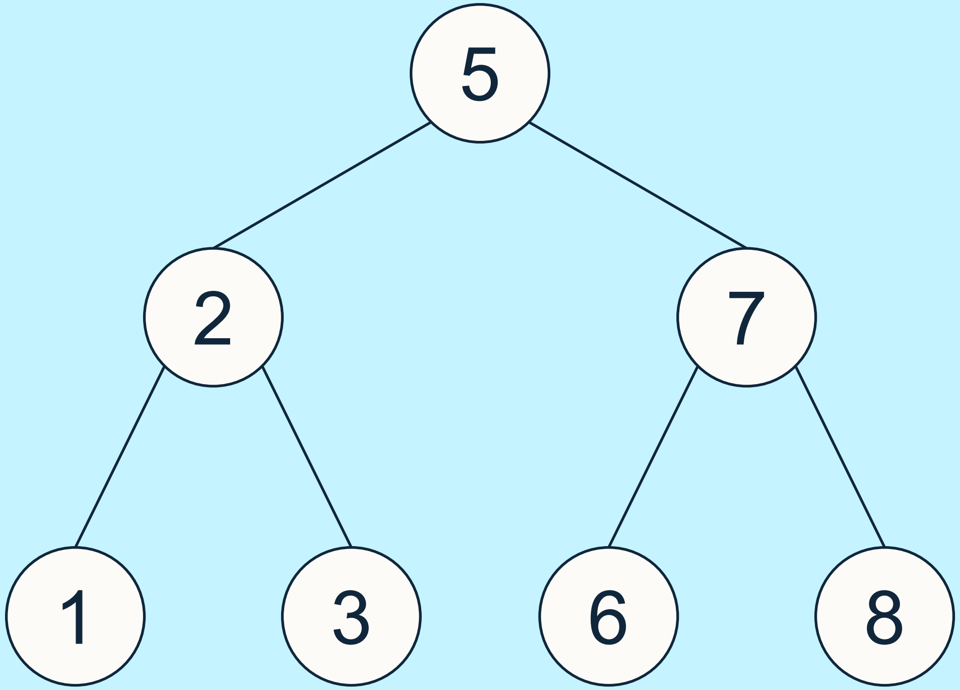
- Starting with the heap:



Perform removeMin() swapping with smallest child (if less than current).

Operations on BSTs – insert(x)

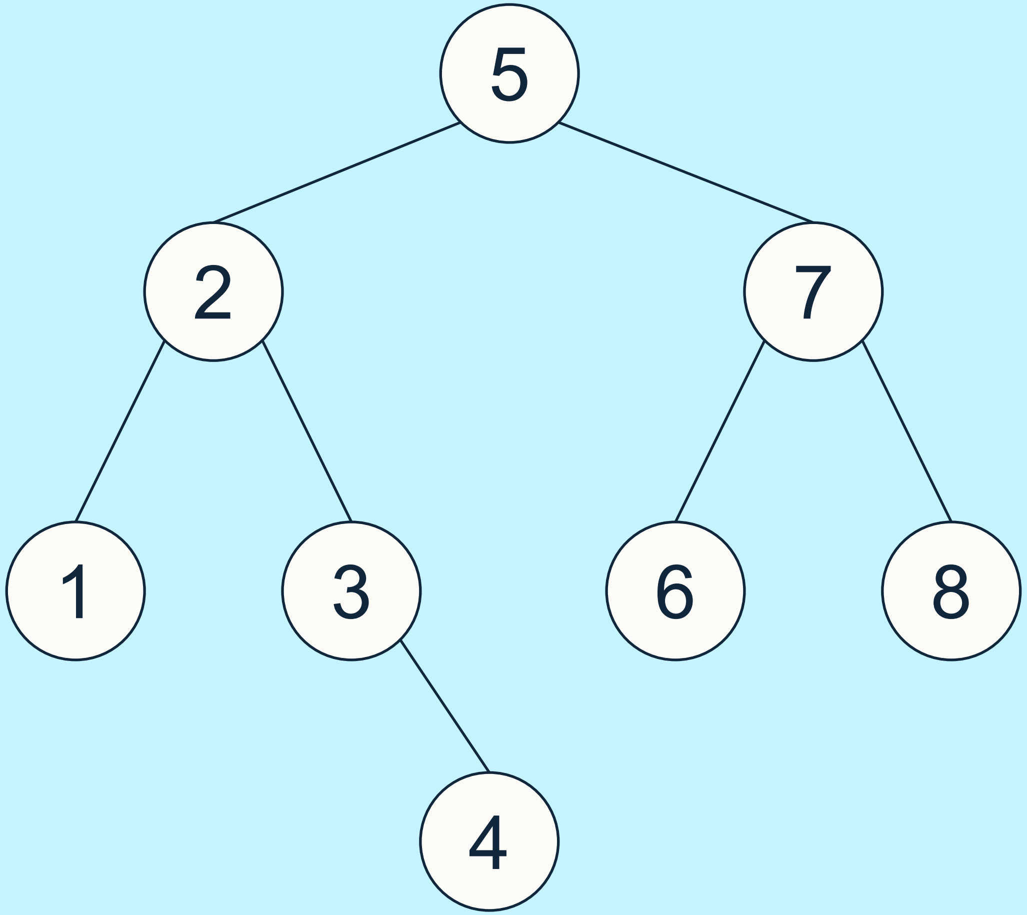
- Starting with the BST:



Perform insert(4).

Operations on BSTs – remove(x)

- Starting with the BST:



Perform remove(2), remove(7), then remove(5).

Questions

Draw the following as binary trees and label each with "Heap", "BST", or "Neither": $T_1 = [-, 1,2,4,7,8,5,6]$ and $T_2 = [-, 1,2,5,7,8,4,6]$.

If you identify any of these as a **heap**, then perform `x = removeMin()` followed by `insertItem(x)`.

- Is the resulting heap identical?

Use heapsort to sort the values [2,3,1,4,3',2',5]. Using your resulting sorted list, explain if heapsort is stable or not.

With the following BST $[-, 6,2,10,1,5,7,11, -, -, 3, -, -, 9, -, -]$ perform at least the following operations and state the array-based representation after each step:

- Remove(7); remove(2); add(7); remove(11); remove(10).

University of Nottingham
UK | CHINA | MALAYSIA

Hash Maps

Array-based Hash Maps with Linear Probing

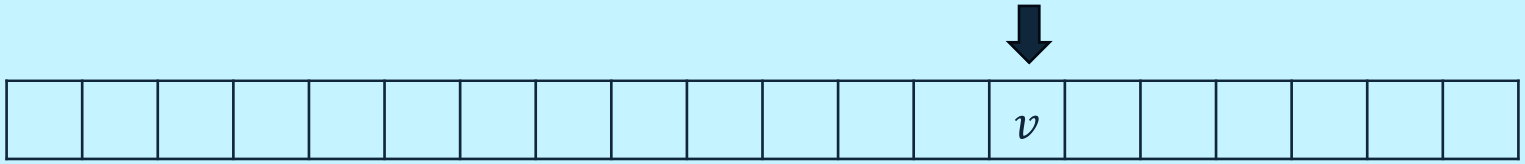
Hash maps

Hashing function to improve average case time complexity of operations from $O(\log n)$ for BST to $O(1)$.

Both BST and hash maps have a worst case of $O(n)$.

Worst case of BST can be improved with self-balancing (e.g. see https://www.geeksforgeeks.org/self-balancing-binary-search-trees/)

$$(k, v) \rightarrow h(k) \rightarrow index$$



A good hash function reduces collisions on the input keys (dispersal)…

…but is unavoidable so need a mechanism to handle this.

Hash maps – Collision Handling

Can use separate chaining or a method of **open addressing** such as linear probing (today’s focus).

Assuming a hash function $h(x) = x \,\%\, n$ where $n = 8$.

Insert the following values (in order): $\{2,4,8,16,32,64\}$

Hash maps – Collision Handling

Can use separate chaining or a method of **open addressing** such as linear probing (today's focus).

Assuming a hash function $h(x) = x \,\%\, n$ where $n = 8$.

Insert the following values (in order): $\{2,4,8,16,32,64\}$

8	16	2	32	4	64		

To do the get(k) with linear probing, need to continue scanning until the key is found, an empty cell is found, or we have inspected all cells.

Hash maps – Collision Handling

Can use separate chaining or a method of **open addressing** such as linear probing (today's focus).

Assuming a hash function $h(x) = x \% n$ where $n = 8$.

Insert the following values (in order): $\{2,4,8,16,32,64\}$

8	16	2	32	4	64		

High number of collisions in both put(k,v) and get(k). How might we redesign the hash function if we know the pattern of the input keys?

Hash maps – Collision Handling

Can use separate chaining or a method of **open addressing** such as linear probing (today's focus).

Assuming a hash function $h(x) = x \,\%\, n$ where $n = 8$.

Insert the following values (in order): $\{2,4,8,16,32,64\}$

8	16	2	32	4	64		

Need to handle remove(k) using either lazy deletion or reinsertion.

Why is "just removing" a key incorrect?

Questions

Insert the following keys into a hash map using linear probing with the hash function $h(k) = (k + 1)\ \%\ 7$ and $c = 1$ into an array of length 7: $\{4, 5, 11\}$.

Next remove 4 using an appropriate removal scheme.

Then explain how you find the key 11.

Using the below hash map and hash function, remove each of $\{2,8,16,32\}$ using lazy deletion, then again with reinsertion. Which method required the **least comparisons to find each key** and which method required the least comparisons overall (including comparison for the reinsertion)?

$$h(x) = x\ \%\ n \text{ where } n = 8$$

8	16	2	32				

University of Nottingham
UK | CHINA | MALAYSIA

Thank you