THE PRIORITY QUEUE

THE PRIORITY QUEUE

WHEN A CERTAIN ELEMENT IN A COLLECTION HAS THE **HIGHEST WEIGHTAGE OR PRIORITY** - A COMMON USE CASE IS TO PROCESS THAT **FIRST**

THE DATA STRUCTURE YOU USE TO STORE ELEMENTS WHERE THE HIGHEST PRIORITY HAS TO BE PROCESSED FIRST CAN BE CALLED A **PRIORITY QUEUE**

AT EVERY STEP WE ACCESS THE ELEMENT WITH THE HIGHEST PRIORITY

THE DATA STRUCTURE NEEDS TO **UNDERSTAND THE PRIORITIES** OF THE ELEMENTS IT HOLDS

THE PRIORITY QUEUE

COMMON OPERATIONS ON A PRIORITY QUEUE:

INSERT ELEMENTS (ALONG WITH PRIORITY INFORMATION)

ACCESS THE HIGHEST PRIORITY ELEMENT

REMOVE THE HIGHEST PRIORITY ELEMENT

PRIORITY QUEUES HAVE A WHOLE NUMBER OF PRACTICAL USE CASES IN EVENT SIMULATION, THREAD SCHEDULING - REAL WORLD PROBLEMS LIKE HANDLING EMERGENCY ROOM CASES ETC.

SO HOW WOULD YOU IMPLEMENT A PRIORITY QUEUE?

LET'S CONSIDER SOME CHOICES

COMMON OPERATIONS ARE INSERT, ACCESS HIGHEST PRIORITY ELEMENT AND REMOVE HIGHEST PRIORITY ELEMENT

AN ARRAY OR A LIST

UNORDERED		ORDERED
CAN BE ANYWHERE IN THE LIST OR ARRAY - COMPLEXITY O(1)	INSERTION	REQUIRES FINDING THE RIGHT POSITION FOR THE ELEMENT BASED ON PRIORITY - COMPLEXITY O(N)
ACCESSING THE HIGHEST PRIORITY ELEMENT REQUIRES GOING THROUGH ALL ELEMENTS IN THE LIST - COMPLEXITY O(N)	ACCESS	ACCESSING THE HIGHEST PRIORITY ELEMENT IS THEN EASY - O(1)
REMOVING THE HIGHEST PRIORITY ELEMENT REQUIRES GOING THROUGH ALL ELEMENTS IN THE LIST - COMPLEXITY O(N)	REMOVE	REMOVING THE HIGHEST PRIORITY ELEMENT IS STRAIGHTFORWARD - COMPLEXITY O(1)

BALANCED BINARY SEARCH TREE

INSERTION

FOR A BALANCED BST THE WORST CASE IS- COMPLEXITY O(LG N)

ACCESS

ACCESSING THE HIGHEST PRIORITY ELEMENT IS AGAIN O(LG N)

REMOVE

REMOVING THE HIGHEST PRIORITY ELEMENT IS O(LG N)

THIS SOLUTION TRADES OF BY MAKING **BOTH INSERTION AND ACCESS MODERATELY FAST** - LIST SOLUTIONS MAKE ONE OF THESE SUPER FAST WHILE COMPROMISING HEAVILY ON THE OTHER

CAN WE DO BETTER?

YES WE CAN!

THE BINARY HEAP

THE BINARY HEAP

INSERTION

INSERTING A NEW ELEMENT - COMPLEXITY O(LG N)

ACCESS

ACCESSING THE HIGHEST PRIORITY ELEMENT IS FAST - O(1)

REMOVE

REMOVING THE HIGHEST PRIORITY ELEMENT IS O(LG N)