

Priority Queue ADT

Min Priority Queue ADT

state

Set of comparable values
- Ordered based on "priority"

behavior

insert(value) – add a new element to the collection.

removeMin() – returns the element with the smallest priority, removes it from the collection.

peekMin() – find, but do not remove the element with the smallest priority.

Uses:

- Operating System
- Well-designed printers
- Some Compression Schemes (google Huffman Codes)
- Sorting
- Graph algorithms

3

3

Even More Operations

BuildHeap(elements e_1, \dots, e_n) – Given n elements, create a heap containing exactly those n elements.

Try 1: Just call insert n times.

Worst case running time?

n calls, each worst case $\Theta(\log n)$. So it's $\Theta(n \log n)$ right?

That proof isn't valid.

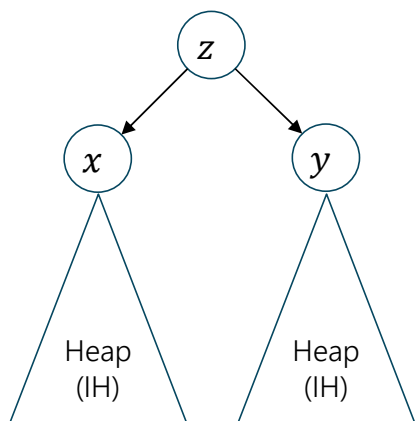
There are at least two distinct problems (bugs or gaps that need much more explanation), can you find them?

18

18

Let's Prove It!

Well, let's sketch the proof of it.



31

31

Amortization

AMORTIZED

It costs \$1800/month (which we pay once)

So the cost per day is $\frac{1800}{30} = 60$.

Good answer if the question is "what does my daily pay need to be to afford housing?"

UNAMORTIZED

On the first it costs \$1800.

Every other day of the month it costs \$0

Good answer if the question is "how much do I need to keep in my bank account so it doesn't get overdrawn?"

38

38