CS 2230
CS II: Data structures

Meeting 31: Priority queue
Brandon Myers
University of Iowa
Today’s big ideas

• A new kind of queue ADT, a **Priority Queue**, where you can remove its min (or max) item

• Priority queues are useful when order depends on rank not just arrival time

• Priority queues are efficiently implemented using (max/min) heaps

• A heap can be implemented as a linked binary tree or an array-based binary tree with the **heap-order property**
Preparation before you searched this morning

1. A "web crawler" finds new or modified pages and puts them into a Map<String,List<String>> where
   • keys are words found in the page
   • values are the list of URLs where that word is found
   (dog, [www.petco.com, en.wikipedia.org/wiki/Dog])
   (iowa, [uiowa.edu, en.wikipedia.org/wiki/Iowa, azdailysun.com/.../iowa/article...])
High level view of a web search engine

Preparation before you searched this morning

1. A "web crawler" finds new or modified pages and puts them into a Map<String,List<String>> where
   • keys are words found in the page
   • values are the list of URLs where that word is found
   (dog, [www.petco.com, en.wikipedia.org/wiki/Dog])
   (Iowa, [uiowa.edu, en.wikipedia.org/wiki/Iowa, azdailysun.com/.../iowa/article...])

2. A base ranking of each page is calculated based on features like
   • how relevant the website is for that search term
   • the quality of the website
   • ...
   List of ranked pages could instead be a “priority queue” that sorts pages by rank
   (leash, [1 www.petco.com, 1 amazon.com, 5 https://en.wikipedia.org/wiki/Leash, ...])
   (Iowa, [1 uiowa.edu, 1 en.wikipedia.org/wiki/Iowa, ...,944 azdailysun.com/.../iowa/article, ...])
When you search

lookup key “iowa”

(In a real search engine, the combination of words is vitally important to ranking. For example, Wikipedia’s rank for “iowa dogs” will be much lower than its rankings for the individual words “iowa” and “dog”.)
Priority queue applications

Situations where the order an element enters the queue doesn’t determine the order it leaves, due to priorities

<table>
<thead>
<tr>
<th>scenario</th>
<th>element</th>
<th>enter queue when</th>
<th>priority determined by</th>
</tr>
</thead>
<tbody>
<tr>
<td>web crawler in a search engine</td>
<td>a web page</td>
<td>web crawler finds the new or changed page</td>
<td>relevance to keyword, quality of site, ...</td>
</tr>
<tr>
<td>?</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

give an example scenario (limit your submitted answer to this column)

fill in the other 3 columns for your scenario

https://b.socrative.com/login/student/
CS2230A ids 1000-4999
CS2230B ids 5000+
Priority queue applications

Situations where the order an element enters the queue doesn’t determine the order it leaves, due to priorities

<table>
<thead>
<tr>
<th>scenario</th>
<th>element</th>
<th>enter queue when</th>
<th>priority determined by</th>
</tr>
</thead>
<tbody>
<tr>
<td>web crawler in a search engine</td>
<td>a web page</td>
<td>web crawler finds the new or changed page</td>
<td>relevance to keyword, quality of site, ...</td>
</tr>
<tr>
<td>air traffic control</td>
<td>airplane</td>
<td>the flying airplane gets close to the airport</td>
<td>late flights go first, planes low on fuel go first, ...</td>
</tr>
<tr>
<td>UI course waitlist</td>
<td>student</td>
<td>register for a class</td>
<td>class level, whether you are in the major, ...</td>
</tr>
</tbody>
</table>
An ADT for priority queues

```java
interface PriorityQueue<P extends Comparable<P>, V> {
    // adds value with the given priority
    void insert (P priority, V value);

    // returns and removes the value with minimum priority
    V deleteMin();

    // returns the value with the minimum key
    V min();

    int size(); // return # of entries
    boolean isEmpty(); // return true if empty
}
```
Peer instruction

Use an array that stores elements in **arbitrary order**. What is the insert time? What is the deleteMin time?

| 10 | 14 | 4 | 15 | 7 | 21 |

**insert, deleteMin**

a) $O(1)$ amortized, $O(\log n)$
b) $O(\log n)$, $O(\log n)$
c) $O(n)$, $O(1)$
d) $O(n)$, $O(n)$
e) $O(1)$ amortized, $O(n)$

[https://b.socrative.com/login/student/](https://b.socrative.com/login/student/)

CS2230A ids 1000-4999
CS2230B ids 5000+
Peer instruction

Use an array that stores elements in **sorted order** in a linked list. What is the insert time? What is the deleteMin time?

```
2 -> 7 -> 12
```

insert, deleteMin

a) \(O(1), O(\log n)\)
b) \(O(n), O(\log n)\)
c) \(O(n), O(1)\)
d) \(O(\log n), O(1)\)
e) \(O(n), O(n)\)

[Link to Socrative](https://b.socrative.com/login/student/)

CS2230A ids 1000-4999
CS2230B ids 5000+
binary heap

another representation for a priority queue is a binary tree filled top to bottom that has the

heap property

• for a min heap: Key at a node ≥ Key at its parent
• for a max heap: Key at a node ≤ Key at its parent

example of a binary min heap
Which of the following is a valid binary max heap?

a) ![Diagram a]
b) ![Diagram b]
c) ![Diagram c]
d) ![Diagram d]

Peer instruction
https://b.socrative.com/login/student/
CS2230A ids 1000-4999
CS2230B ids 5000+
Array representation of a binary heap

for parent at index $k$, child nodes are stored at
$left = 2k$
$right = 2k + 1$
$parent = \lfloor k/2 \rfloor$

we are leaving index 0 empty...why?
Inserting into a heap

put it in the next available spot
bubble up the inserted node until its parent is smaller than it
Deleting from a binary heap

Before:
- Tree with values 1, 4, 2, 9, 7, 13, 5, 22, 12, 8

After deleteMin operation:
- Key 1 returned
- Tree simplified with value 8 moved to root
Deleting from a binary heap

1. Original heap: 1, 4, 2, 9, 7, 13, 5, 22, 12, 8
2. After deleteMin: 4, 9, 7, 13, 5, 22, 12, 8
3. Remove key 1 and move 8 to the root:
   - 8 moved to root
   - Key 1 returned
4. Final heap: 2, 4, 5, 9, 7, 13, 8, 22, 12
Peer instruction

Using a binary min heap for our Priority queue, what is the insert and deleteMin time?

insert, deleteMin

a) $O(1), O(\log n)$
b) $O(\log n), O(1)$
c) $O(n), O(1)$
d) $O(\log n), O(\log n)$
e) $O(n), O(n)$

https://b.socrative.com/login/student/
CS2230A ids 1000-4999
CS2230B ids 5000+
Today’s big ideas

• A new kind of queue, a **Priority Queue**, that polls its items in sorted order regardless of the order they were added.

• Applications of priority queues: sorting, paging search engine results, picking the top K scores.

• Priority queues are efficiently implemented using *(max/min) heaps*.

• A heap can be implemented as a linked binary tree or an array-based binary tree.
resources

• animations of heaps

http://www.cs.usfca.edu/~galles/visualization/Heap.html
acknowledgements

heap insert/deleteMin diagrams
http://homepage.divms.uiowa.edu/~ghosh/2116.8.pdf