Priority Queues

Algorithms & Data Structures
ITCS 6114/8114

Dr. Dewan Tanvir Ahmed
Department of Computer Science
University of North Carolina at Charlotte

Priority Queue ADT

- A priority queue stores a collection of items
- An item is a pair (key, element)
- Main methods of the Priority Queue ADT
  - insertItem(k, o)
    inserts an item with key k and element o
  - removeMin()
    removes the item with smallest key and returns its element
- Additional methods
  - minKey()
    returns, but does not remove, the smallest key of an item
  - minElement()
    returns, but does not remove, the element of an item with smallest key
  - size(), isEmpty()
- Applications:
  - Standby flyers
  - Auctions
  - Stock market

Total Order Relation

- Keys in a priority queue can be arbitrary objects on which an order is defined
- Two distinct items in a priority queue can have the same key
- Mathematical concept of total order (linear order, simple order, or (non-strict) ordering) relation $\leq$
  - Reflexive property:

$$x \leq x$$

  - Antisymmetric property:

$$x \leq y \wedge y \leq x \Rightarrow x = y$$

  - Transitive property:

$$x \leq y \wedge y \leq z \Rightarrow x \leq z$$

Sorting with a Priority Queue

- We can use a priority queue to sort a set of comparable elements
  1. Insert the elements one by one with a series of insertItem(e, e) operations
  2. Remove the elements in sorted order with a series of removeMin() operations
- The running time of this sorting method depends on the priority queue implementation

```
Algorithm PQ-Sort(S, C)
    Input sequence S, comparator C
    for the elements of S
    Output sequence S sorted  in
    increasing order according to C
    P ← priority queue with
        comparator C
    while ¬S.isEmpty ()
        e ← S.remove (S. first ())
        P.insertItem(e, e)
    while ¬P.isEmpty()
        e ← P.removeMin()
        S.insertLast(e)
```

Rearrange elements in increasing order

Or at least nondecreasing order if there are ties

Sequence-based Priority Queue

- Implementation with an unsorted sequence
  - Store the items of the priority queue in a list-based sequence, in arbitrary order
- Performance:
  - insertItem takes O(1) time since we can insert the item at the beginning or end of the sequence
  - removeMin, minKey and minElement take O(n) time since we have to traverse the entire sequence to find the smallest key

- Implementation with a sorted sequence
  - Store the items of the priority queue in a sequence, sorted by key
- Performance:
  - insertItem takes O(n) time since we have to find the place where to insert the item
  - removeMin, minKey and minElement take O(1) time since the smallest key is at the beginning of the sequence

Reference

- Algorithm Design: Foundations, Analysis, and Internet Examples. Michael T. Goodrich and Roberto Tamassia. John Wiley & Sons.
- Introduction to Algorithms. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein.