EECS 2011 M: Fundamentals of Data Structures

Suprakash Datta
Office: LAS 3043

Course page: `http://www.eecs.yorku.ca/course/2011M`
Also on Moodle

Priority Queues

Ch. 9.1, 9.2, 9.4,9.5
Applications

- job scheduling shared computing resources (Unix)
- Event simulation
- As a building block for other algorithms

Note: Some slides in this presentation have been adapted from the authors' slides.

Priority Queue ADT

set S of elements, each with an associated *key* value
Operations supported:

- Insert(S, x) insert element x in set S ($S \leftarrow S \cup \{x\}$)
- Maximum(S) returns the element of S with the largest key
- Extract-Max(S) returns and removes the element of S with the largest key
- Others: Change priority(x) changes the key of element x, Max(S), size(S), isEmpty(S)

Priority Queue - Trace

Method	Return Value	Priority Queue Contents
insert(5,A)		{ (5,A) }
insert(9,C)		{ (5,A), (9,C) }
insert(3,B)		{ (3,B), (5,A), (9,C) }
min( )	(3,B)	{ (3,B), (5,A), (9,C) }
removeMin( )	(3,B)	{ (5,A), (9,C) }
insert(7,D)		{ (5,A), (7,D), (9,C) }
removeMin( )	(5,A)	{ (7,D), (9,C) }
removeMin( )	(7,D)	{ (9,C) }
removeMin( )	(9,C)	{ }
removeMin( )	null	{ }
isEmpty( )	true	{ }

Priority Queue - Keys

- Keys must have total order relationship (comparability, anti-symmetry, transitivity)
- Different elements can have the same key
- Key Interface:

```
public interface Entry<K,V> {
    K getKey(); //returns the key for this entry
    V getValue(); //returns the value associated with
                  //this entry
}
```

- Comparator ADT: $compare(x, y)$ returns an integer < 0 if $x < y$, returns 0 if $x = y$, and an integer > 0 if $x > y$

Priority Queue - Example

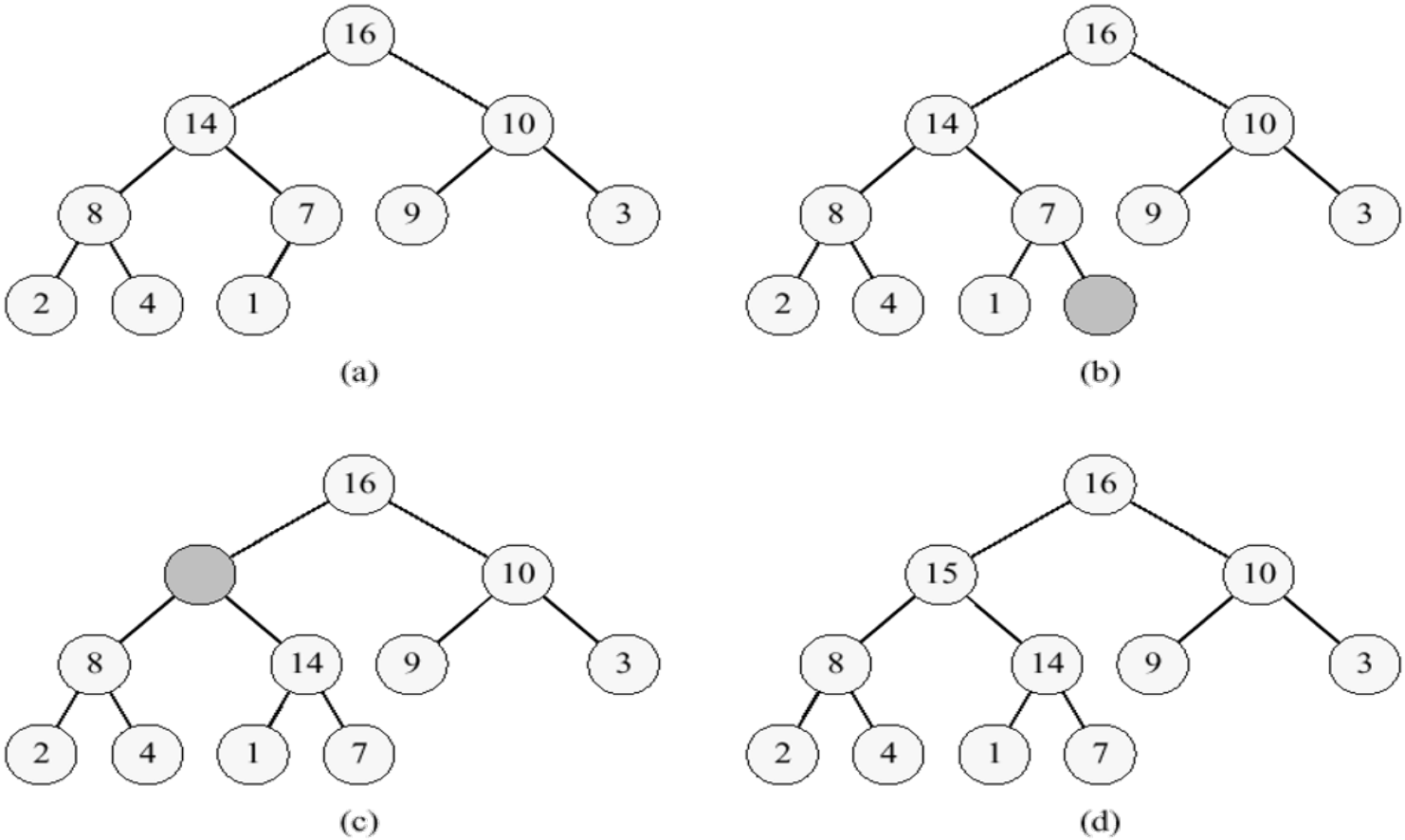
```
/** Comparator for 2D points under lexicographic order */
public class Lexicographic implements Comparator {
  int xa, ya, xb, yb;
  public int compare(Object a, Object b) throws
                ClassCastException {
     xa = ((Point2D) a).getX();
     ya = ((Point2D) a).getY();
     xb = ((Point2D) b).getX();
     yb = ((Point2D) b).getY();
     if (xa != xb)
         return (xb - xa);
     else
         return (yb - ya);
   }
}
```

Priority Queue - Example contd.

```
/** Class for a point in the plane with integer coords */
public class Point2D {
  protected int xc, yc; // coordinates
  public Point2D(int x, int y) {
        xc = x;
        yc = y;
  }
  public int getX(){
      return xc;
    }
  public int getY() {
      return yc;
    }
}
```

Priority Queue: Heap Implementation

Insertion:



Extract-Max: one step of heap sort

Priority Queue: other implementations

1. A sorted array
   - Insertion - linear time
   - Extract min - constant time
   - Change priority - linear time
2. A linked list
   - Insertion - linear time
   - Extract min - constant time
   - Change priority - linear time