Algorithms about heaps

(1) Sort :

```
while ( siz > 1 )
{
    swap ( heap, 1, siz )
    siz -- ;
    siftDown( heap, 1 )
}
```

(2) delete / dequeu (Serve) :

```
e = heap [1]
heap [1] = heap [siz]
siz --
siftDown (heap, 1)
```

(3) insert / enqueu

```
siz = siz + 1
heap [siz] = e
siftup ( heap, siz)
```

```
public boolean isHeap( int x [], int siz)
{
      int i = 1;
      while ( x [i] < x [2*i] && x [i] < x[2*i+1] &&
              i < siz/2 )
              i ++;
      if ( i <= siz/2)
            retrn false
      else
            return true;
}
```

Q4 : priority Queue as heap, lower Value heigher p

Serve one time (dequeue)

	5	7	6	9	11	13	17	10	14
	1	2	3	4	5	6	7	8	9

	14	7	6	9	11	13	17	10	
	1	2	3	4	5	6	7	8	

i = 1
Left = 2
Right = 3

	6	7	14	9	11	13	17	10
	1	2	3	4	5	6	7	8

i = 3
Left = 6
Right = 7

	6	7	13	9	11	14	17	10
	1	2	3	4	5	6	7	8

Enqueue 8 :

	6	7	13	9	11	14	17	10	8
	1	2	3	4	5	6	7	8	9

i = 9
parent = i/2 = 4

	6	7	13	8	11	14	17	10	9
	1	2	3	4	5	6	7	8	9

i = 4
parent = 2