p3q3
MR

1 Data structures and algorithms

2003

(1) In the first phase of heapsort, an initially random vector is rearranged to satisfy the heap structure constraints. Describe what these are, how the rearrangement is done, and carefully prove that it can be done in $O(n)$ time, where n is the number of elements in the vector. [8 marks]

(2) Complete the description of heapsort and show that its worst case performance is $O(n)$. [8 marks]

(3) How many element comparisons would your implementation use to sort the integers 1 to 8 if they were (a) initially in sorted order, and (b) if they were initially in revesre sorted order. [4 marks]

```
ANSWER NOTES:

Heapsort is a standard sorting method covered in the course.

(1)
Children of node i (if any) are at 2i and 2i+1.
Constraints: if 2i<=n   v[i] >= v[2i]
             if 2i+1<=n v[i] >= v[2i+1]

FOR i = n/2 TO 1 BY -1 DO downheap(v, i, n)
where downheap should be described.

Cost:
worst cost of downheap = 2 x depth of its tree

For a heap of depth d there are N = 2**(d+1)-1 nodes.
                                  ~ 2 x 2**d
```

d	nodes at level d	i	downheap cost	
0	1	d	2 x 1	x d
1	2	d-1	2 x 2	x (d-1)
2	4	d-2	2 x 4	x (d-2)
k	2**k	i	2 x 2**(d-i)	x i
d-1	2*(d-1)	1	2 x 2**(d-1)	x 1

```
                     TOTAL = 2 x sigma(i = 1 to d)(2**(d-i) x i
                           = 2 x 2**d x sigma(i=1 to d) (i/(2**i))
                           ~ N        x K
                                since sigma(i=1 to d) (i/(2**i)) converges.
(2) Bookwork for the algorithm
    Cost ~ N x cost of downheap
         ~ N x 2 x d
         ~ 2N (lg N -1)
         = O(N lg N)

(3a)                                   cost in compares
1 23 4567 8    downheap(v, 4, 8)    1    (48)
1 23 8567 4    downheap(v, 3, 8)    2    (36 37)
1 27 8563 4    downheap(v, 2, 8)    3    (28 25 24)
1 87 4563 2    downheap(v, 1, 8)    4    (18 17 14 15)
8 57 4163 2                                        (end of phase1)
```

```
2 57 4163    8 downheap(v, 1, 7)   4   (25 27 26 23)
7 56 4123    8
3 56 412    78 downheap(v, 1, 6)   3   (35 36 32)
6 53 412    78
2 53 41    678 downheap(v, 1, 5)   3   (25 23 24)
5 43 21    678
1 43 2    5678 downheap(v, 1, 4)   3   (14 13 12)
4 23 1    5678
1 23     45678 downheap(v, 1, 3)   2   (12 13)
3 21     45678
1 2     345678 downheap(v, 1, 2)   1   (12)
2 1     345678
1      2345678 done
                     TOTAL COST = 26 comparisons

8 76 5432 1    downheap(v, 4, 8)   1   (51)
8 76 5432 1    downheap(v, 3, 8)   2   (63 62)
8 76 5432 1    downheap(v, 2, 8)   2   (75 74)
8 76 5432 1    downheap(v, 1, 8)   2   (87 86)
8 76 5432 1                                       (end of phase1)
1 76 5432    8 downheap(v, 1, 7)   4   (17 16 15 14)
7 56 1432    8
2 56 143    78 downheap(v, 1, 6)   3   (25 26 23)
6 53 142    78
2 53 14    678 downheap(v, 1, 5)   4   (25 23 21 24)
5 43 12    678
2 43 1    5678 downheap(v, 1, 4)   3   (24 23 21)
4 23 1    5678
1 23     45678 downheap(v, 1, 3)   2   (12 13)
3 21     45678
1 2     345678 downheap(v, 1, 2)   1   (12)
2 1     345678
1      2345678 done
                     TOTAL COST = 24 comparisons
```