Make-up Exam01 2004

1.

For find(x) and remove(x) operations, it is easy to show that the amortized cost is O(1).

For put(x) operations,

(i) when doubling does not occur,
the actual cost of put(x) operation is o(1)

(ii) when doubling occurs,
the actual cost is the half of the capacity of the structure prior to the operation is performed.

Thus sum(actual cost of n put(x) operations) is

n + { sum(2**i) 1<=i<=k, where k = ceil(log n) }

= n + 2*(2**k -1) - k
<= 5n -2 - k
< 5n

Therefore, "5" is enough for the amortized cost of put(x) operation.

2.

S=180 records
n=810 records
m runs
ts = 8 ms
tl = 2 ms
tt = 0.1 ms / record

(a)

(1) b = floor(S / (2k+2)) = floor (180/ (2k+2))

(2) time to read a buffer
= (8 + 2 + (1)*0.1) = (10 + (1)*0.1) ms

(3) # of buffers per pass
= ceiling(n / b) = ceiling(810 / (1))

(4) input time per pass
= (2) * (3)

(5) # of passes
= ceiling(log m base k)

(6) total input time
= (4) * (5)

	k=2	k=8
(1)	180/6 = 30	180 / 18 = 10
(2)	10+30*0.1 = 13 ms	10+10*0.1 = 11 ms
(3)	810/30 = 27	810/10 = 81
(4)	13*27 = 351 ms	11*81 = 891 ms
(5)	log m base 2	log m base 8
(6)	351* log m base 2	891 * log m base 8

Thus, k=8 is better

(b) Total input time = 891 * log m base 8
Total output time = 891 * log m base 8
Total merge time = (810*0.1) * log m base 8

Thus (2*891 + 81) * log m base 8 ms

3.

```
                (4,90)
            /              \
       (14,70)            (20,84)
      /        \          /      \
  (30,61)   (24,43)   (31,71)   (80,83)

                (4,90)
            /              \
       (14,70)            (20,84)
      /        \          /      \
  (30,61)   (24,43)   (31,71)   (80,83)
 /
(95)

                (4,90)
            /              \
       (14,70)            (20,84)
      /        \          /      \
  (30,95)   (24,43)   (31,71)   (80,83)
 /
(61)
```

```
              (4,90)
           /              \
       (14,95)            (20,84)
     /        \          /       \
  (30,70)    (24,43)    (31,71)    (80,83)
  /
(61)
```

```
                (4,95)
           /              \
       (14,90)            (20,84)
     /        \          /       \
  (30,70)    (24,43)    (31,71)    (80,83)
  /
(61)
```

Part (b)

```
                 (61,95)
           /              \
       (14,90)            (20,84)
     /        \          /       \
  (30,70)    (24,43)    (31,71)    (80,83)
```

```
                (14,95)
           /              \
       (61,90)            (20,84)
     /        \          /       \
  (30,70)    (24,43)    (31,71)    (80,83)
```

```
                (14,95)
           /              \
       (24,90)            (20,84)
     /        \          /       \
  (30,70)    (61,43)    (31,71)    (80,83)
```

```
                  (14,95)
           /              \
       (24,90)            (20,84)
     /        \          /       \
  (30,70)    (43,61)    (31,71)    (80,83)
```

4.

(a) left tree is already a min leftist tree
    right tree is converted the below tree

```
              2
             / \
            /   \
          50     7
         / \     /
        /   \   /
       80   70 11
       /
      /
     90
```

(b)

Step 1.

```
                 7
                / \
               /    \
             11      12
                     /
                    /
                   14
          / \
             /   \
                15    17
```

Step2

```
          6
        /   \
       /     \
          10      7
         /       / \
        /       /   \
       12      11    12
      /              /
     /              /
    30             14
          / \
             /   \
                90    17
```

swap the left and right subtree at rooted 6

```
          6
        /   \
```

```
       /     \
          7       10
         /\        /
        /  \      /
       11  12     12
            /     /
          /     /
         14    30
        / \
       /   \
      90   17
```

Step4 : this is a last step of the meld two trees

```
          2(3)
       /       \
      /         \
     50(2)       6(2)
    / \         /    \
   /   \       /      \
 80(1) 70(1) 7(2)       10(1)
  \           / \         /
   \         /   \       /
  90(1)    11(1) 12(1) 12(1)
                /      /
               /      /
            14(2)   30(1)
            / \
           /   \
         90(1)  17(1)
```

5.

one of the possible Solutions)

Step1 :

After Delete-Min

```
4    3       33         18      16    15
     |      / |        /  |      |
    21    34  45      19  21     60
          |           |
          47          26
```

Pairwise combine after delete-Min

(Combine two heaps with degree 0)

```
4     3       33         18      16
|     |      / |       /  |     |
15    21   34  45    19   21   60
               |     |
               47    26
```

(Combine two heaps with degree 1)

```
   16     3           33        18
   |    /  |         / |       /  |
   60  4   21      34  45     19  21
       |               |      |
       15              47     26
```

(combine two heaps with degree 2)

```
           3          33    16
       /   /  |      /  |     |
      18  4   21   34  45    60
     / |  |             |
    19 21 15            47
    |
    26
```