1. amortized time complexity.

charge 2 to the ball on the put(x) operation.
one for putting x into the jar and
one for pulling x out of the jar.

Therefore the amortized cost is O(1).

2. external sorting.

(a) need one dummy run (length zero).

First merge 4 runs (0, 0, 10, 20) into one (30 records)
And merge the rest and new one (20,30,40,60)
into one (150 records)

(b)
need two comparisons to generate one record, except for the first record of each merge step due to loser tree initialization.
(3 comparisons to initialize the loser tree)

For the first merge step, # of comparisons = 2 * (30 - 1) + 3
For the second merge step, # of comparisons = 2 * (150 - 1) + 3

So, the total # of comparisons =61 + 301 = 362

(c) Due to sequential processing,
```
merge time = Input_time + Output_time + CPU time
           = 2 * Input_time + CPU time
```

```
So, [2(3*2) + 3] + [2(15*2) + 15]
    = 15 + 75 = 90 (sec)
```

3. interval heap (omitted steps)

(a) insert 90,

```
                5,95
             /       \
        15,90          25,85
        /    \        /    \
     35,75  25,45  35,75  81,82
     /
```

```
      65
```

(b) DeleteMin :

- pull out 5
- re-insert 81, swapping if necessary.

```
                 15,95
               /       \
          25,81          25,85
          /    \       /    \
       35,65  45,75  35,75  82
```

4.

* Note) the number in parentheses is shortest(x) value of node x.

(a)

```
               1(2)
              /  \
             /    \
            5(2)  4(1)
            / \
           /   \
          8(2)  7(1)
         / \    /
        /   \  /
      41   30  9
      (1)  (1) (1)
```

(b)

```
               1(3)
              _/  \_
            _/      \_
           /          \
         4(2)         3(2)
        _/ \_        _/ \_
```

```
       /     \      /     \
      8(1)  9(1)   5(2)   7(1)
            /      / \
           /      /   \
          15    6(2)  13(1)
         (1)    / \
               /   \
              10   11
             (1)   (1)
```

5. min bionomial heap (omitted steps)

(a) insert keys

```
                             min
   -6--2--5--13--10--8--3--9--1--12--4-
```

(b) DeleteMin

- delete 1
- merge each min bionomial heaps with single element
  (Using degree table of size O(logN))

```
    min
     2               4
   / | \             |
  6  5  3           12
     |  | \
    13  9  8
           |
           10
```