Question 1.

a. Aggregate method. There are $\lfloor \sqrt{n} \rfloor$ operations that each cost $4\sqrt{k}$, and the other $n - \sqrt{\lfloor n \rfloor}$ operations each cost 1. This yields a total cost of $2 * (1 + 2 + 3 + ... + \sqrt{\lfloor n \rfloor}) + (n - \sqrt{\lfloor n \rfloor}) * 1 = 2* ((\sqrt{\lfloor n \rfloor})* (\sqrt{\lfloor n \rfloor} + 1) / 2) + (n - \sqrt{\lfloor n \rfloor}) =< n + \sqrt{\lfloor n \rfloor} + n - \sqrt{\lfloor n \rfloor} = 2n$. So the amortized cost is $2n / n = 2$.

b. Accounting method. Charge each operation \$1. When k is not a perfect square, use \$1 to pay for the operation and use the extra \$1 for credit. When k is a perfect square, the preceding $(k - 1) - (\sqrt{k} - 1)^2 = 2\sqrt{k} - 2$ operations have each paid a credit of \$1, which together with the \$2 for the current operation yields exactly enough to pay for its $2\sqrt{k}$ actual cost.

c. Potential method. Define Φ = the number of operations since the most recent perfect square. That is, let $\Phi_K = k - (\sqrt{\lfloor k \rfloor})^2$. When k is not a perfect square, the amortized cost is $c' = c + \Phi_k - \Phi_{k-1} = 1 + (k - (\sqrt{\lfloor k \rfloor})^2) - ((k - 1) - (\sqrt{\lfloor k \rfloor})^2) = 2$. When k is a perfect square, the amortized cost is $c' = c + \Phi_k - \Phi_{k-1} = 2\sqrt{k} + (k - (\sqrt{k})^2) - ((k - 1) - (\sqrt{k} - 1)^2) = 2\sqrt{k} + 0 - (2\sqrt{k} - 2) = 2$.

Question 2.

(a) m=5, n=4
# of dummy nodes needed
(1-m) mod (n-1) = (1-5) mod (4-1) = -4 mode 3 = (6-4) mod 3 = 2
Pass 1: 4 runs with 0, 0, 500, 700 records ==> 1200-record run
Pass 2: 4 runs with 1200, 900, 1100, 1500 records ==> 4700-record run

(b) total time = I/O time + Merge time
Pass 1: 2*1200*10 / 100 + 12 = 252 seconds
Pass 2: 2*4700*10 / 100 + 47 = 987 seconds
Thus, total 1239 seconds

Question 3.

(a)

Step 1. Adjust end points in each node if needed

```
                    (39,70)
                 /          \
          (82,99)            (1,12)
         /        \          /    \
     (20,23)     (20,48)  (49,63) (35,50)
     /     \       /
(19,25)(50,57) (19,46)
```

Step 2-1. Starting from the parent of the last node, from right to left and from bottom to top.

```
                         (39,70)
                      /          \
                (82,99)           (1,12)
              /         \          /    \
          (19,57)      (19,48)  (49,63) (35,50)
          /     \       /
     (20,25)(23,50) (20,46)
```

Step 2-2.

```
                        (39,70)
                      /          \
                (19,99)           (1,63)
              /         \          /    \
          (20,82)      (19,48)  (12,49) (35,50)
          /     \       /
     (25,57)(23,50) (20,46)
```

Step 2-3 and Final.

```
                        (1,99)
                      /          \
                (19,82)          (12,63)
              /         \          /    \
          (20,70)      (19,48)  (39,49) (35,50)
          /     \       /
     (25,57)(23,50) (20,46)
```

(b)

i) remove the min element from the root

ii) remove the left pointer (20) from the last node and reinsert it into the root

```
                        (20,99)
                      /          \
                (19,82)          (12,63)
              /         \          /    \
          (20,70)      (19,48)  (39,49) (35,50)
          /     \       /
      (25,57)(23,50) (46)
```

==>

```
                        (12,99)
                      /          \
                (19,82)          (20,63)
              /         \          /    \
          (20,70)      (19,48)  (39,49) (35,50)
          /     \       /
      (25,57)(23,50) (46)
```

Question 4.

“ Meld right subtree of tree with smaller root and all of other tree.”

```
          3
       /     \                                        4                            swap
     6      meld[ 8,   4 ]        ==>               /   \              ==>    5     ==>   5
   /    \               /  \                      7    meld[ 5, 8]              \           /
 9      11            7     5                   /                                 8         8
                     /                        10
                   10

                4                                 3
    ===>      /    \         ===>              /       \
             7       5                        6          4
            /       /                       /  \       /   \
          10      8                        9    11    7     5
                                                      /     /
                                                    10    8
```

Question 5.

a) Before DeleteMin: () points to the min element

```
     7    12    4   22   (3)    24    13    23    19    5
```

b) After Deletemin:

```
           ( 4 )          5
          /  |  \
        22  7   13
        |    |  \
       12   24 19
                |
                23
```

c)

$$| B_k | = | B_0 | + | B_1 | + ... + | B_{k-1} |$$

$$= 1 + 2 + ... + 2^{k-1}$$

$$= 2^k$$

$$| B_k | = | B_{k-1} | + | B_{k-1} |$$

$$= 2^{k-1} + 2^{k-1}$$

$$= 2^k$$