NOTE1: This is the solution set for exam 2.

NOTE2: To keep you up to date on class scores, the homework and exam averages follow (for local students). For NTU/FEEDS students, the HWs and exams come in at different times from different students, hence your statistics cannot be meaningful until all work has been completed.

For Local Students Only

	AVG	STD
HW 1	32	7.4
HW 2	39	10.5
EX 1	37	4.6
EX 2	37	7.8

1. (10) Consider a min binomial heap with the following elements:

   `3, 6, 4, 2, 16, 22, 15, 34, 9, 14.`

   Each element defines a min tree of the binomial heap. Perform a DeleteMin operation and show the resulting min trees.

```
      min
       |
       |
       V
      (3)---------14
     / | \
    6  4  9
       |  | \
      16 34 15
             |
            22
```

2. (10) Follow the 2-3 tree algorithms discussed in the class/text.

```
       4,6
      / | \
    1,3 5  7
```

   (a) (5) Draw the resulting 2-3 tree after inserting 2.

```
        4
       / \
      2   6
     / \ / \
    1  3 5  7
```

   (b) (5) Draw all possible trees resulting from the operation delete 6 (seperate from part a, ie. on the original tree not the tree resulting from part a).

```
      3,5
     / | \
    1  4  7

       4
      / \
    1,3 5,7
```

3. (10) Use the bottom-up (2-pass) style algoritms of Red-Black Trees for this question.

   (a) (5) Insert the keys 9, 8, 7, 2, 6, 1, 3 and 4 into an initially empty red-black tree in the order shown, showing the resulting tree following each insertion.

```
         6
      //   \\
     //     \\
    //       \\
   2           8
  / \         / \
 /   \       /   \
1     3     7     9
      \\
       \\
        4
```

   (b) (5) Draw the figures for RRr and RLr color changes and RRb, RLb rotations for the insert operation in a red-black tree.

You can find the LL mirror images in our book, pages 596-597.

```
o RRr                              o RLr

    |             ||                   |             ||
    g              g                   g              g
  //  \\  ==>   /   \                // \\   ==>    /   \
 s    p        s     p               s   p         s     p
      \\             \\                 //               //
        q              q                q                q
```

```
o RRb                              o RLb

     |              |                   |              |
     g              p                   g              q
   / \\  ==>    //   \\               /  \\  ==>   //   \\
  s   p         g     q              s    p        g     p
     / \\      / \   / \                // \      / \   / \
    a   q     s   a b   c               q   c    s   a b   c
       / \                             / \
      b   c                           a   b
```

4. (10)

   (a) (5) A single 2-3-4 tree can be represented by many red-black trees. How many red-black trees represent a 2-3-4 tree with x 2-nodes, y 3-nodes, and z 4-nodes (where x, y and z are variables/numbers, an example is x=1, y=2, and z=4)?

   Answer: 2^y

   (b) (5) What is the number of failure/external nodes in this red-black tree?

   Answer: $= 1 + x + 2y + 3z$

5. (10) Using the optimal binary search tree algorithm discussed in class, compute r_{ij}, and c_{ij}, $0 \leq \mathrm{i} < \mathrm{j} \leq 3$ for the identifier set $(a_1, a_2, a_3) =$ (do, for, while) with $p_1 = 3/10$, $p_2 = 1/5$, $p_3 = 5/20$, $q_0 = 1/20$, $q_1 = 1/20$, $q_2 = 1/10$, $q_3 = 1/20$. Using the r_{ij}'s, construct the optimal binary search tree.

```
( a1, a2, a3 ) = ( do, for, while )
multiply by 20 to make simple
p1=3/10=6, p2=4, p3=5
q0=1/20=1, q1=1/20=1, q2=1/10=2, q3=1/20=1

    w00 = q0, w11 = q1, w22 = q2, w33 = q4
    c00 = c11 = c22 = c33 = 0
    r00 = r11 = r22 = r33 = 0

    w01 = q0 + q1 + p1 = 1 + 1 + 6 = 8
    c01 = w01 + min{ c00 + c11 } = 8
    r01 = 1

    w12 = q1 + q2 + p2 = 1 + 2 + 4 = 7
    c12 = w12 + min{ c11 + c22 } = 7
    r12 = 2

    w23 = q2 + q3 + p3 = 2 + 1 + 5 = 8
    c23 = w23 + min{ c22 + c33 } = 8
    r23 = 3

    w02 = q0+q1+q2+p1+p2 = 1 + 1 + 2 + 6 + 4 = 14
    c02 = w02 + min{ c00+c12, c01+c22 } = 14 + min{7,8} = 21
    r02 = 1

    w13 = q1+q2+q3+p2+p3 = 1 + 2 + 1 + 4 + 5 = 13
    c13 = w13 + min{ c11+c23, c12+c33 } = 13 + min{8,7} = 20
    r13 = 3

    w03 = q0+q1+q2+q3+p1+p2+p3 = 20
```

```
c03 = w03 + min{c00+c13, c01+c23, c02+c33} = 20 + min{21,16,20} = 36
r03 = 2

OBST:    r03=2                    for
          /   \                 /   \
     r01=1   r23=3    ===>    do   while
```