This is an open text and open notes exam.

Write your answers on your own paper. The instructor will have blank paper in case you have none.

1. Suppose that we have a school with \( n \) students. These students have student numbers \( 0, 1, 2, \ldots, n - 1 \). The teachers of the school will vote for the “most outstanding student” award, and there are \( \sqrt{n} \) teachers.

Describe how to determine which of the \( n \) students received the largest number of the \( \sqrt{n} \) votes (if two or more students tie for the highest number of votes, you can just pick one winner arbitrarily). The votes are given to you in an array \( V[1 \ldots \lceil \sqrt{n} \rceil] \), and for each \( i \), \( 0 \leq V[i] < n \).

For full credit, tabulate the votes in time \( O(\sqrt{n}) \). [10 points]

2. Now for another voting problem. There are \( n \) students, but here the students are voting for the “best teacher award”. The votes are given in an array \( V[1 \ldots n] \), where \( V[i] \) is the ID number of the teacher (which can be a very very large number, unlike the previous problem).

Furthermore, we insist that the winner of this award receive an absolute majority of the votes (\( \geq \lceil \frac{n}{2} + 1 \rceil \) of them).

Let \( n \) be odd, and the median value of \( V \) to be

\[
\mu = \text{RandomizedSelect}(V, 1, n, \lceil \frac{n + 1}{2} \rceil)
\]

(a) How long does it take to find \( \mu \) using \text{RandomizedSelect}, on average and in the worst case? (Do not give the algorithm - it’s in the text - just the time bounds.)

(b) It is possible that there is no majority element in \( V \). Argue that if there is a majority element, then it is \( \mu \).

(c) Use the above to derive an \( O(n) \) (expected time) algorithm to determine if \( V \) has a someone receiving a majority of the votes, and, if so, return that value (teacher ID, here).

[10 points]

3. Insert the following values into an initially empty B-tree with parameter \( t = 3 \):

\[
5, 9, 18, 37, 2, 7, 10, 15, 22, 30, 3, 16, 17, 23, 13, 11
\]

As in the text, there will be at least \( t = 3 \) children (at least \( t - 1 = 2 \) keys) per node and at most \( 2t = 6 \) children (\( 2t - 1 = 5 \) keys). [11 points]
4. Start with the Fibonacci heap of figure 1 at the end of the exam. This is a min heap - it happens to consist of just one tree.
   
   (a) Decrease 35 to 15.
   (b) Then decrease 50 to 20.
   (c) Insert 16, 17, 18
   (d) Perform an extractMin.

[10 points]

5. We are given a BST $T$ and an integer $k$, and want the values stored in nodes at level $k$, from left to right. (The fields of each node are $val$, $lchild$, and $rchild$.) That is, we want $p.val$ for each node $p$ in $T$ of depth $k$. Provide an $O(n)$ algorithm to provide this list of values. Why is it $O(n)$? [10 points]

6. Regarding red-black trees
   
   (a) Into an initially empty RB tree insert the values: 41, 38, 31,12, 19, 8, 15, 5, 10, 6, 4, 2.
   (b) From the RB tree of figure 2 (dotted lines mean red), delete 11.

[12 points]

7. Recall the problem on homework 6 about the water jugs: we have $n$ red water jugs and $n$ blue water jugs, all different shapes and sizes. Also, for each red water jug, there is a unique blue water jug holding the same amount of water. A comparison here consists of pouring water from a jug of one color into a jug of another color (you have an unlimited supply of water). You can tell as a result of this comparison if one jug is smaller than the other, larger than the other, or exactly the same size.

Describe an $O(n)$ randomized algorithm to find the 4th smallest red water jug. [10 points]

Total: 73 points
Figure 1: Fibonacci heap for question 4

Figure 2: red-black tree for question 6