1. Consider the use of lemma 8.4 of the text (p 199) to sort \( n \) numbers of \( b \) bits each. Suppose we choose \( b = \lg n \lg \lg n \).

   (a) What is the largest integer that can be expressed (in unsigned binary) using \( b \) bits?
   (b) If \( r \) is chosen to be \( \lg n \), how long does RADIUSORT take, according to the lemma?
   (c) How long does it take if \( r = \lg \lg n \) is chosen instead?
   (d) Which choice of \( r \) is faster?

   [10 points]

2. Suppose \( A \) is an array of \( n \) integers, and consider the following piece of code that constructs a binary search tree from \( A \):

   ```
   buildBST(A)
   
   BST T
   for i = 1 to n
       T.insert(A[i])
   return T
   ```

   Let \( C \) be the total number of comparisons made by `buildBST`, and let \( I \) be the internal path length (p 1180) of the tree \( T \) returned by `buildBST`. Compare \( C \) and \( I \). Is \( C \) bigger than \( I \), smaller, unrelated? Justify your answer. [10 points]

3. Put the following values into an initially empty B-tree of different parameters

   \[15, 6, 12, 30, 22, 17, 10, 5, 9, 14, 23, 1, 17, 29, 18, 24, 7, 26, 35, 33, 27\]

   (a) parameter \( t = 3 \), which will have at least 3 children (at least 2 keys) per node and at most 6 children (at most 5 keys).
   (b) parameter \( t = 4 \)

   [14 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. Marked nodes are circled.

   (a) Decrease 35 to 29.
(b) Then decrease 22 to 13.
(c) And then decrease 49 to 24.
(d) Insert 21, 22, 23
(e) Perform an extractMin.

[14 points]

5. Regarding red-black trees

(a) From the RB tree of figure 2 (dotted lines mean red), delete 30.
(b) Again from the RB tree of figure 2 (the original one, before part (a)), delete 1.

[12 points]

6. Suppose you have an array $S$ of size $n$, where each element of the array represents a vote for a person. A vote $S[i]$ is represented by the SSN of the candidate (so it can be a very large number). We do not know the number of candidates. A person wins if they receive a majority (at least $\lceil \frac{n+1}{2} \rceil$) of the votes. Give an efficient procedure to determine if there is a winner (and, if so, who). How fast is your method? [10 points]

**Total:** 70 points

Figure 1: Fibonacci heap for question 4
Figure 2: red-black tree for question 5