CSE326 Assignment #1

Due: Friday, July 5, 2002

1. For each of the following sets of constraints below, say whether or not a function $f(n)$ exists which meets the constraints. If a function exists, give an example of such a function.

   a) $f(n) \in o(n^2)$ and $f(n) \in \Omega(n)$

   b) $f(n) \in O(n^3)$ and $f(n) \in \Theta(n^3)$ and $f(n) \in \Omega(n^3)$

   c) $f(n) \in \Omega(n \log n)$ and $f(n) \in \Theta(n)$

   d) $f(n) \in \Omega(1)$ and $f(n) \in o(\log(n))$

   e) $f(n) \in o(n^2)$ and $f(n) \in \Theta(n^2)$

   f) $f(n) \in \omega(n)$ and $f(n) \in O(n)$

   g) $f(n) \in \omega(\log(n))$ and $f(n) \in o(n)$

   h) $f(n) \in \omega(1)$ and $f(n) \in o(\log(n))$

   i) $f(n) \in o(1)$

2.

   a) Give an algorithm to determine if a positive integer, N, is prime.

   b) In terms of N, what is the worst case running time of your program? (You should be able to do this in $O(\sqrt{N})$.)

   c) Let B equal the number of bits in the binary representation of N. What is the value of B?

   d) In terms of B, what is the worst-case running time of your program?

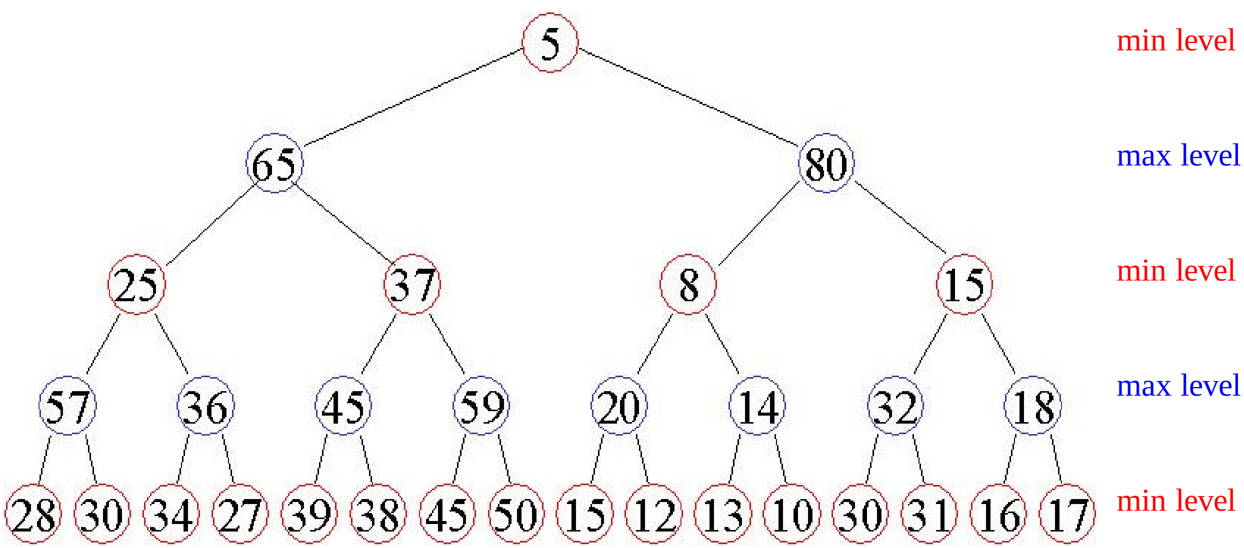
   e) Compare the running times to determine if a 20-bit number and a 40-bit number are prime.

   f) Is it more reasonable to give the running time in terms of N or B? Why?

3. For a binay heap stored in an array, the parent of node i is stored in position $\lfloor i/2 \rfloor$, the left child is in position $2i$, and the right child is in position $2i+1$. What about a d-heap stored in an array? In what positions are the children and parent of node i stored?

4. Consider the worst case for the **deleteMin** method for binary heaps which we looked at in class (and in the textbook). When percolating down the empty node, we require up to $\log(N)$ comparisons to determine minimum children nodes and up to $\log(N)$ comparisons to determine whether the last node in the heap can be safely inserted into the empty node (so the heap order property is satisfied). This is a total of approximately $2\log(N)$ comparisons. Describe an algorithm that only requires approximately $\log(N) + \log(\log(N))$ comparisons in the worst case for this operation.

5. The picutre below is an example of a *min-max* heap, a data structure used to implement a double-ended priority queue. A *min-max* heap is a variation on the normal binary heap data structure. Like a normal binary heap, a *min-max* heap has the structure property that it must be a complete binary tree. However, the heap order property is a bit different for *min-max* heaps: every node at an even depth in the tree is smaller than its parent but larger than its grandparent, and every node at an odd depth in the tree is larger than its parent but smaller than its grandparent. A *min-max* heap supports the **insert**, **deleteMin**, and **deleteMax** operations in $O(\log(N))$ time.



a) Where is the node with minimum priority located in the tree? Where is the node with maximum priority located?

b) <u>EXTRA-CREDIT</u> Give algorithms for the *min-max* heap **deleteMin** and **deleteMax** operations.