CS2223

MIDTERM EXAM

Name_____

Date: November 18, 2004
All documentation permitted

1	_____
2	_____
3	_____
4	_____
TOTAL	_____

1. (25 points) Suppose that $A[1..n]$ is a sorted list of distinct numbers, and $B[1..n]$ is a sorted list of distinct numbers. That is, $A[i] < A[i+1]$ and $B[i] < B[i+1]$ for $1 \le i < n$. We want to know if the lists are disjoint. That is, we want to know if there are $i, j, 1 \le i, j \le n$ such that $A[i]=B[j]$. Show a $O(n)$ upper bound on the complexity of testing if A and B are disjoint.

2. (25 points) Suppose you are given a sorted list $A[1..n]$ of $n>1$ distinct integers, and a pair of integers *lo* and *hi* satisfying $A[1] \le lo \le hi \le A[n]$. Describe an algorithm to return the interval of A whose values fall within *lo* and *hi*. That is, if your algorithm should return i and j, $1 \le i \le j \le n$ satisfying

- $A[k] < lo$ for $k<i$,
- $lo \le A[k] \le hi$ for $i \le k \le j$, and
- $hi<A[k]$ for $k>j$.

Your algorithm should have a worst-case execution time in $O(\lg n)$.

3. (25 points) For each of the bottom five rows and for each of the rightmost three columns, check all of the columns which apply. For example, if $\frac{n^2-12n}{4} \in O(65n)$, then check the 3^{rd} column of the 2^{nd} row.

$f(n)$	$g(n)$	$f(n) \in O(g(n))$	$g(n) \in O(f(n))$	$f(n) \in \Theta(g(n))$
$\frac{n^2-12n}{4}$	$65n$			
$n\sqrt{n}/2$	$n \lg n$			
2^n	3^n			
$42n^2\sqrt{n}$	$n^2\left(\sqrt{n}+\ln n\right)$			
$n!$	4^n			

4. (25 points) Give pseudocode for an algorithm TESTIFHEAP? which will test if array $A[1..n]$ is a (min-)heap using, in the worst-case, $O(n)$ operations. You may assume that n is odd.

CS2223

Solutions to Midterm Exam

1. MERGE($A[1..n]$,$B[1..n]$) $\Rightarrow$ $C[1..2n]$ $\Theta(n)$

   disjoint? $\leftarrow$ true $O(1)$

   for $i \leftarrow 1$ **to** $2n$-1 **do**

   if $C[i]=C[i+1]$ **then** *disjoint?* $\leftarrow$ false $\Theta(n)$

   return *disjoint?* $O(1)$

2. Do a BINARYSEARCH of *lo* in *A*, $O(\lg n)$
   returning either *i* such that $A[i]=lo$ or the
   index *i* of the smallest element of *A* greater than *lo*.
   Do a BINARYSEARCH of *hi* in *A*, $O(\lg n)$
   returning either *j* such that $A[j]=hi$ or the
   index *j* of the largest element of *A* less than *hi*.

3.

$f(n)$	$g(n)$	$f(n) \in O(g(n))$	$g(n) \in O(f(n))$	$f(n) \in \Theta(g(n))$
$\frac{n^2-12n}{4}$	$65n$		X	
$n\sqrt{n}/2$	$n \lg n$		X	
2^n	3^n	X		
$42n^2\sqrt{n}$	$n^2\left(\sqrt{n}+\ln n\right)$	X	X	X
$n!$	4^n		X	

4. *Heap?* $\leftarrow$ true

   for $i \leftarrow 1$ **to** $\lfloor n/2 \rfloor$ **do**

   if $A[i] > A[2i]$ **or** $A[i] > A[2i+1]$ **then** *Heap?* $\leftarrow$ false

   return *heap?*