CS 466/666
Assignment 3
Fall 2005
Due date: Friday November 11st, 6:00pm

1. Compute the smallest proof of intersection for the sets

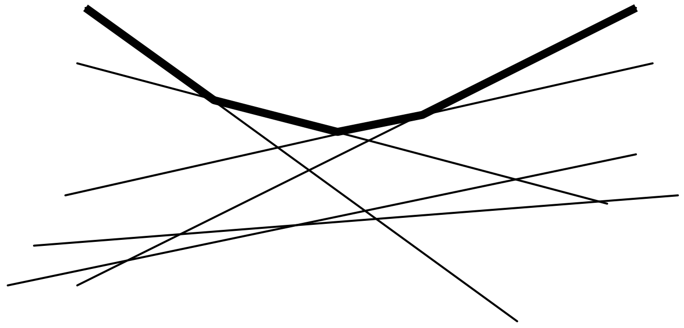
A_1	26 27 28 29 30 73 77 78 79 91 196 206 330 352 353 364 365 384 408 426 427
A_2	76 77 145 207 216 366 388 389 390 402 417 421
A_3	21 24 25 26 27 28 29 77 119 120 199 298 339 350 389 408

Here's the greedy algorithm for computing the smallest proof of intersection

```
e ← $\max_i$ { $A_i$[1] }
while ( e < infinity )
  tmax ← -infinity
  for i from 1 to k do
    find the insertion rank $p_i$ of e in set $A_i$ i.e. $A_i[p_{i-1}]$ < e <= $A_i[p_i]$
    if tmax < $A_i[p_i]$ then
      tmax ← $A_i[p_i]$
      j ← i
    endif
  endfor
  if $A_j[p_j]$ = e  (i.e. all sets contain e) then
    add the k-1 equalities $A_1[p_1]=A_2[p_2]=...=A_{k-1}[p_{k-1}]=A_k[p_k]$ to the proof
    e ← $\max_i$ { $A_i[p_{i+1}]$ }
  else
    add $A_j[p_j]$ > e to the proof
    e ← $A_j[p_j]$
  endif
endwhile
```

2. In computer graphics one of the most important problems is efficiently computing which surfaces are hidden in a 3D collection of objects being rendered, either because the are part of the "back side" of an object or because another object occludes it (i.e. blocks the view).

For this question we consider a simplification of this problem to 2D as follows. Given n non-vertical lines in the plane, labeled $L_1, L_2, \ldots, L_{n-1}, L_n$ with the ith line specified by the equation $y = ax_i + b_i$, we wish to find the line segments that would be visible from a "fly over" high above. More formally we say line L_i is uppermost at a given x-coordinate u if its y-coordinate v at that point is greater than the y coordinates of all the other lines at that x-coordinate u. We say line L_i is visible if there is some x-coordinate at which it is uppermost–intuitively, some portion of it can be seen if you look down from "$y = \infty$" .



i. Give a divide and conquer $O(n \log n)$ algorithm for computing the subsegments of the lines that can be seen from infinity.
ii. Assume that only h of the n lines are visible from infinity and the remaining n-h lines are completely occluded, give an algorithm that computes the visible subsegments in $O(n \log h)$ time. (*Hint: first divide the lines into groups of k for some value of k and compute the visible lines within each group).*

3. Show that computing the maximum, median and minimum elements in a given a set of n distinct elements in random order requires at least $2n$ comparisons in the worst case.

4. Give an algorithm for the problem above requiring as few comparisons as possible, in the worst case (*Hint: aim for 9n/4 comparisons*).

5. Given an array of n integer values in random order, we wish to determine if it is an arithmetic progression when viewed in sorted order. An arithmetic progression is defined to be a set of the form $\{a, a+b, a+2b, \ldots ,a+(n-1)b\}$ for some integers a and b. Give an algorithm to solve this problem in $O(n)$ time.

6. Given n integers in unsorted order, consider the problem of determining if they are all different or if there is a repeated element. Show that this takes $O(n \log n)$ time in the worst case in the comparison model. *(Hint: first show that if the relative order of each element with respect to every other is not known then an adversary can make the algorithm fail).*