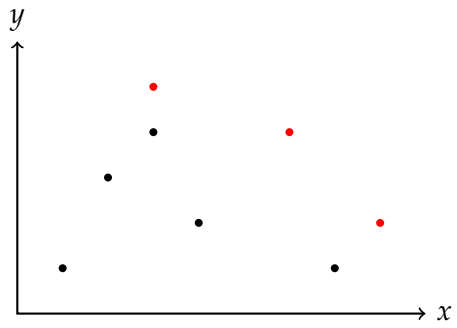
Due Date: September 9, 11:59pm

Problem 1: [10pts] An *inversion* in an array $A[1...n]$ is a pair of indices (i, j) such that $i < j$ and $A[i] > A[j]$. Describe and analyze an algorithm that counts the number of inversions in an n-element array in $O(n \log n)$ time. (**Hint:** *Modify mergesort.*)

Problem 2: [10pts] Let $S = \{p_i = (x_i, y_i) : 1 \leq i \leq n\}$ be a set of n points on a 2D plane. We say that a point p_i *dominates* p_j if $x_i \geq x_j$ and $y_i \geq y_j$. A point p_i is a maximal point of S if there are no other points in S dominating p_i. In the example below, the red points are maximal points of S. Describe and analyze an algorithm that returns the maximal points in $O(n \log n)$ time.
(**Hint:** *Divide S into two sets based on the x-coordinates of points. What is the merge step?*)



Problem 3: [10pts] Recall that the algorithm **LazySelect** we see in class performs $2n + o(n)$ comparisons with probability at least $1 - O(n^{-\frac{1}{4}})$. Proved that the expected number of comparisons can be improved to $1.5n + o(n)$ by modifying the algorithm.