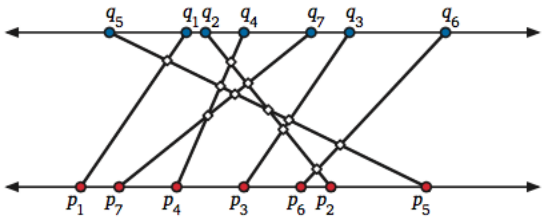
CS180 Homework 2

Due: 4:00pm, 04/13/2016

1. You are given n songs $\{1, 2, \ldots, n\}$ and 2 preference lists $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_n\}$. Each a_i is a positive integer and $a_i = k$ means the song k is ranked at i^{th} position in list A. An inversion between A and B is a pair (i, j) such that song i is ranked before song j in list A, but song j is ranked before song i in list B. In the lecture, we showed that to count the number of inversions, we can permute the songs in both lists according to the index of list A. More specifically, the permutation is defined by $a_i \mapsto i$ for all i. We claimed that the number of inversions after applying the permutation does not change. Actually, this claim works for any permutation. Prove that the number of inversions does not change if we apply an *arbitrary* permutation.

2. Suppose you are given two sets of n points in the plane, one set $\{p_1, p_2, ..., p_n\}$ on the line $y = 0$ and the other set $\{q_1, q_2, ..., q_n\}$ on the line $y = 1$. Create a set of n line segments by connecting each point p_i to the corresponding point q_i. The number of intersections is the number of how many pairs of these line segments are crossing.



An example with 11 intersections

   (a) Consider the two sets of n points are two preference lists of n songs. The value of p_i and q_i represents how song i is ranked in two preference lists. Prove that the number of inversions in two preference lists is equivalent to the number of intersections.

   (b) Describe an algorithm to calculate the number of intersections in $O(n \log n)$ time.

3. A celebrity among n persons is someone who is known by everyone but does not know anyone. Give an iterative algorithm to find the celebrity among n people.

4. The diameter of a tree is the number of edges in the longest path in the tree.

   (a) Given a rooted directed tree, find the diameter of the underlying undirected tree using a recursion, computes the result of the original problem given that we removed the root and solved the problem for the remaining rooted trees.

   (b) Write the iterative version of this algorithm, and analyze its complexity, which should be linear in n, the number of nodes in the tree.

★ Homework assignments are **STRICTLY** due on the exact time indicated. Please submit a hard copy of your homework solution with your **Name**, **Bruin ID**, **Discussion Number**, clearly indicated on the first page. If your homework consists of multiple pages, please **staple** them together. Email attachments or other electronic delivery methods are not acceptable.

★ We recommend to use LaTeX, LyX or other word processing software for submitting the homework. This is not a requirement but it helps us to grade the homework and give feedback. For grading, we will take into account both the correctness and the clarity. Your answer are supposed to be in a simple and understandable manner. Sloppy answers are expected to receive fewer points.

★ Unless specified, you should justify your algorithm with proof of correctness and time complexity.