**Nuts and bolts.** A disorganized carpenter has a mixed pile of *n* nuts and *n* bolts. The goal is to find the corresponding pairs of nuts and bolts. Each nut fits exactly one bolt and each bolt fits exactly one nut. By fitting a nut and a bolt together, the carpenter can see which one is bigger (but the carpenter cannot compare two nuts or two bolts directly). Design an algorithm for the problem that uses at most proportional to *n*log*n* compares (probabilistically).

**Selection in two sorted arrays.** Given two sorted arrays *a*[] and *b*[], of lengths *n*1​ and *n*2​ and an integer 0≤*k*<*n*1​+*n*2​, design an algorithm to find a key of rank *k*. The order of growth of the worst case running time of your algorithm should be log*n*, where *n*=*n*1​+*n*2​.

* Version 1: *n*1​=*n*2​ (equal length arrays) and *k*=*n*/2 (median).
* Version 2: *k*=*n*/2 (median).
* Version 3: no restrictions.

**Decimal dominants.** Given an array with *n* keys, design an algorithm to find all values that occur more than *n*/10 times. The expected running time of your algorithm should be linear.