Algorithms and Data Structures
Spring 2018
Exercises for Units 3 and 4

1. For a leaf v in a binary tree T let $d(v)$ denote the depth of v in T, i.e. the number of edges on the path from the root of T to v. Let $d(T) = \sum_{v \text{ leaf of } T} d(v)$.

   (a) Show that in a binary tree with n leaves there is a leaf v with $d(v) \geq \log_2 n$.

   (b) Prove that in any binary tree T with n leaves $d(T) \geq n\lfloor \log_2 n \rfloor$.

2. We have seen that any algorithm that finds the minimum and maximum of n keys simultaneously requires at least $\lceil \frac{3n}{2} \rceil - 2$ comparisons. Devise an algorithm based on the adversarial argument that uses exactly $\lceil \frac{3n}{2} \rceil - 2$ comparisons.

3. Let $n \geq 3$ be an odd integer. The median of n distinct keys S is defined as the key m with the property that $\lfloor \frac{n}{2} \rfloor$ of the keys in S are smaller than m, and that $\lfloor \frac{n}{2} \rfloor$ of the keys in S are larger than m. Use an adversarial argument to show that at least $\frac{3(n-1)}{2}$ comparisons are required to find the median.

   Hint: Partition the keys into four sets (S) the keys smaller than or equal to the median (L) the keys larger than or equal to the median (M) the median (N) the keys for which nothing is known so far.

4. The so-called "$a + b \neq c$"-problem takes as input three sorted sets A, B, C of $\Theta(n)$ numbers each and asks whether for all $a \in A$, $b \in B$, and $c \in C$ we have $a + b \neq c$.

   We will restrict our attention to algorithms for this problem that employ only comparisons (or, ask queries) of the form $x + y < z$, $x + y > z$, or $x + y = z$ with $x \in A$, $y \in B$, and $z \in C$.

   (a) Show that this problem can be solved using $O(n^2)$ comparisons.

   (b) Use an adversary argument to show that in the worst case we need to ask at least $\Omega(n^2)$ such queries to determine whether indeed no bad triple with $a + b = c$ exists.

   Hint: Consider the three sets $A = \{-10n + 4i | 0 \leq i < n\}$, $B = \{10n + 4i | 0 \leq i < n\}$, and $C = \{4i + 1 | 0 \leq i < 2n\}$. If you pick two numbers a from A and b from B, then there is a unique $c \in C$ that differs only by 1 from $a + b$. What happens if you "perturb" a and b by adding 1/3 to each, and subtracting 1/3 from c? Which queries does this perturbation affect? How can the adversary use this property in order to derive an $\Omega(n^2)$ lower bound?

5. Consider the problem DISJOINTNESS of testing whether two sets $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\}$ of real numbers are disjoint.

   Outline a simple algorithm for solving this problem for which you can show that it has optimal asymptotic worst case running time.

   What is the algorithm and its running time? For which model of computation do you prove worst case optimality? What is your "lower bound proof?"

6. Consider the problem DIAMETER: For a set $P = \{(u_i, v_i) \,|\, 1 \leq i \leq n\}$ you are to decide whether it contains two points that have distance at least 2 between them.

   Here is an argument that seems to show that DIAMETER is at least as difficult as DISJOINTNESS: Given an instance A, B of DISJOINTNESS we transform it in linear time to an instance P_{AB} of DIAMETER so that the answer to the latter problem is YES iff the answer to the original problem is NO. I.e., P_{AB} contains two points that are at least distance 2 apart iff the sets A and B are not disjoint. Here is how this transformation works: W.l.o.g. assume all reals in A and in B lie in the open interval $(-1, 1)$. (What would you do enforce this assumption?) Now let

   $$P_{AB} = \{(\cos a_i, \sin a_i)|1 \leq i \leq n\} \cup \{(\cos(\pi + b_j), \sin(\pi + b_j))|1 \leq j \leq n\}\,.$$

   In effect this places the points of A on the "right half" of the unit circle and B on the "left half" in such a way that if A and B have a number in common the corresponging points will be diametrically opposed on the unit circle and the distance between them will be 2.

   (a) This argument is fallacious in that it cannot be used to deduce a lower bound for DIAMETER in the algebraic computation tree model. What is wrong with it?

   (b) Fix the argument. (Hint: "rational parametrization")