

CS 580 HW 6

Ankur Dhoot

April 3, 2017

Q1

We'll use an adversary similar to the one used in the element uniqueness problem.

We'll group the n numbers in our set $X = \{v_1, \dots, v_n\}$ into groups of 2 and 3 subject to minimizing the absolute difference between the number of groups of 2 and the number of groups of 3.

Since $n > 6$, we are guaranteed to have at least one group of 3 and one group of 2 in such a grouping. For example: $n = 7 = 2 + 3 + 2$ or $n = 8 = 3 + 2 + 3$ or $n = 9 = 2 + 3 + 2 + 2$ or $n = 10 = 3 + 2 + 3 + 2$ or $n = 11 = 3 + 2 + 3 + 3$.

It's not difficult to see that the groupings can be made such that there are at least $\lfloor n/6 \rfloor$ groups of 2 and at least $\lfloor n/6 \rfloor$ groups of 3.

The adversary will assume that both numbers in a group of 2 are equal and all three numbers in a group of 3 are equal. We'll prove a lower bound on the number of comparisons required between numbers in different groups, which is obviously then a lower bound on the total number of comparisons any algorithm must do.

All numbers will lie in the interval $(0,1]$. Let $(a_i, b_i]$ denote the interval in which the numbers in the i th group can lie. Initially, $(a_i, b_i] = (0,1]$ for all groups.

The adversary will do the following when the algorithm asks for a comparison between v_i and v_j :

If v_i and v_j are in the same group, return that $v_i = v_j$.

Else, let $m_i = \frac{a_i + b_i}{2}$ and let $m_j = \frac{a_j + b_j}{2}$ where $(a_i, b_i]$ is the interval of the group to which v_i belongs and $(a_j, b_j]$ is the interval of the group to which v_j belongs.

If $m_i \leq m_j$, then let $b_i = m_i$ and let $a_j = m_j$. Answer $v_i < v_j$.

Else, let $a_i = m_i$ and let $b_j = m_j$. Answer $v_i > v_j$.

Thus, whenever we compare numbers in different groups, the intervals corresponding to each group is halved.

Claim: If the algorithm halts before all groups of 2 have pairwise non-overlapping intervals or before all groups of 3 have pairwise non-overlapping intervals, then the algorithm is not correct.

Proof: Suppose the algorithm halts, but there exist a pair of overlapping intervals corresponding to groups of 2. Let T = algorithm answers that every distinct value in X occurs a number of times that is prime and there are at least two different multiplicities of the numbers in X . Let F = algorithm answers otherwise.

If the algorithm returns T , consider one of the pair of overlapping intervals, $(a_i, b_i]$ and $(a_j, b_j]$ corresponding to elements $\{v_{i_1}, v_{i_2}\}$ and $\{v_{j_1}, v_{j_2}\}$, respectively. If, we let $v_{i_1} = v_{i_2} = v_{j_1} = v_{j_2}$, and let all other groups have distinct numbers (i.e the numbers inside each group are the same, as assumed, but no two groups have the same numbers), then we're left with a bunch of numbers that occur exactly two times, a bunch of numbers that occur exactly three times, and one number that occurs exactly four times (since $v_{i_1} = v_{i_2} = v_{j_1} = v_{j_2}$). But four is not a prime number, so the algorithm has failed.

If the algorithm returns F , then we can just let ALL groups have distinct numbers. We are left with a bunch of numbers that occur exactly twice and a bunch of numbers that occur exactly thrice. Since two and three are prime numbers and not equal to each other, the algorithm should have returned T , so the algorithm has failed.

An analogous analysis holds if the algorithm halts before all groups of 3 have pairwise non-overlapping intervals.

Thus, at the end, the groups of 2 have pairwise non-overlapping intervals and the groups of 3 have pairwise non-overlapping intervals. This implies that the sum of the interval lengths of groups of 2 is ≤ 1 as is the sum of the interval lengths of groups of 3.

Letting l_i denote the length of interval $(a_i, b_i]$ at the end, we have that $\sum l_i \leq 2$. Since $l_i = (1/2)^{c_i}$ where c_i is the number of comparisons in which elements of group i were compared to elements of some other group, we have $\sum (1/2)^{c_i} \leq 2$.

Let k denote the number of groups. Since there at least $\lfloor n/6 \rfloor$ groups of 2 and $\lfloor n/6 \rfloor$ groups of 3, $k \geq 2\lfloor n/6 \rfloor$. From the proof in class, $c_1 + c_2 + \dots + c_k$ is minimized subject to the constraint $\sum (1/2)^{c_i} = \text{some constant}$ (in this case, some constant ≤ 2) when $c_1 = c_2 = \dots = c_k$. When $\sum_{i=1}^k c_i$ is minimized, we can deduce that $(1/2)^{c_i} \leq 2/k \Rightarrow c_i \geq \log_2 k/2 \geq \log_2 \lfloor n/6 \rfloor$. Thus, $\sum_{i=1}^k c_i \geq k \log_2 \lfloor n/6 \rfloor \geq 2\lfloor n/6 \rfloor \log_2 \lfloor n/6 \rfloor$ meaning any correct algorithm must make $\Omega(n \log n)$ comparisons.

Q2

We'll reduce the element uniqueness problem to the minimum tri-distance problem in $O(n)$ time. Since we have already established that element uniqueness requires $\Omega(n \log n)$ time in the ACT model, the minimum tri-distance problem must take $\Omega(n \log n)$ time in the ACT model.

Let $X = \{v_1, \dots, v_n\}$ denote a set of n numbers. Let $Y = \{(v_1, 0), \dots, (v_n, 0)\}$ be a set of n points corresponding to X (each y coordinate is set to 0). Let $Z = \{(v_1, 0), (v_1, 0), \dots, (v_n, 0), (v_n, 0)\}$ be the set of $2n$ points where each point in Y has been repeated twice.

Claim: All numbers in X are unique if and only if the minimum tri-distance of points in Z is > 0 .

Proof:

(\Rightarrow) Suppose all the numbers in X are unique. Then, no two points in Y are the same. Thus, every point in Z occurs exactly twice. Since no point in Z occurs three times, the minimum tri-distance must be < 0 .

(\Leftarrow) Suppose the minimum tri-distance of points in Z is > 0 . Then, no three points in Z can be the same, otherwise the minimum tri-distance would be 0. Thus, all points in Y must be unique (Suppose not, then Y must contain at least two of the same points $\Rightarrow Z$ must contain at least four of the same points \Rightarrow minimum tri-distance in Z is 0, a contradiction). If all points in Y are unique, it's obvious that all numbers in X are unique.

Given an input $X = \{v_1, \dots, v_n\}$ to the element uniqueness problem, we can clearly construct $Z = \{(v_1, 0), (v_1, 0), \dots, (v_n, 0), (v_n, 0)\}$ in $O(n)$ time. We'll use Z as the input to the minimum tri-distance problem. Using the above claim, we'll check the minimum tri-distance, δ . If $\delta = 0$, then not all numbers in X are unique. Otherwise, all numbers in X are unique. Thus, once given δ , we can answer the element uniqueness problem in $O(1)$ time.

Thus, since we can create the input to the minimum tri-distance problem in $O(n)$ time and use the answer to the minimum tri-distance to answer element uniqueness in $O(1)$ time, the minimum tri-distance must take $\Omega(n \log n)$ in the ACT model since element uniqueness requires $\Omega(n \log n)$ time in the ACT model.