CS3230: Assignment for Week 1 Solutions

Due: Sunday, 23rd Jan 2022, 11:59:59 pm SGT.

---

1. We claim that $k = 9$.

   Algorithm: For each $i \in \{2, 3, \ldots, 10\}$, check whether $A_1 = A_i$. If the answer is Yes for all 9 comparisons, return Yes. Else, return No. The correctness is clear.

   Adversarial argument: Consider an algorithm that always makes at most 8 comparisons. The adversary always answers Yes on every comparison. Construct a graph G on 10 nodes indexed by $1, 2, \ldots, 10$, where nodes i and j have an edge between them iff A_i and A_j have been compared with each other. Since G has ≤ 8 edges, it is disconnected. Take any node v, and let C be the connected component containing v. Note that C is not the whole graph G.

   One possibility for the array A is that all numbers are equal. Another possibility is that all numbers A_i for $i \in C$ are equal to some integer x, while all numbers A_i for $i \notin C$ are equal to another integer $y \neq x$. The algorithm cannot distinguish between these two possibilities, so it must give a wrong answer for at least one of the possibilities.[1]

2. Solution 1: Let the four numbers be a, b, c, d.

   1. Compare a with b. Assume without loss of generality that $a < b$.
   2. Compare c with d. Assume without loss of generality that $c < d$.
   3. Compare a with c. The smaller of the two numbers is the smallest number overall.
   4. Compare b with d. The larger of the two numbers is the largest number overall.
   5. Compare the remaining two numbers to decide the second and third largest number overall.

   Solution 2: We can perform comparisons in a slightly different way.

   1. Compare a with b, a with c, and b with c to determine the order of a, b, c. Assume without loss of generality that $a < b < c$.
   2. Compare d with b.

[1] More generally, it could be that the numbers in each connected component C_j are equal to some integer x_j, and the x_j's are different for different j's.

- If $d > b$, compare d with c to determine the largest number.
- If $d < b$, compare d with a to determine the smallest number.

Solution 3: From lecture, we know that Merge Sort requires at most $n \lg n - n + 1 = 4 \lg 4 - 4 + 1 = 5$ comparisons.

3. You can proceed as follows:

   1. Divide the 16 coins into two sets of 8 coins each, and weigh the two sets against each other. If they have equal weight, you are done. Else, let S_1 be the lighter subset. Note that both fake coins must be in S_1.
   2. Divide the 8 coins in S_1 into two sets of 4 coins each, and weight the two sets against each other. If they have equal weight, you are done by adding 4 other coins to each set arbitrarily. Else, let S_2 be the lighter subset. Note that both fake coins must be in S_2.
   3. Divide the 4 coins in S_1 into two sets of 2 coins each, and weight the two sets against each other. If they have equal weight, you are done by adding 6 other coins to each set arbitrarily. Else, let S_3 be the lighter subset. You know that S_3 must contain exactly the two fake coins, and you can accomplish the task.