Name: Guoqing Yu

Problem #3

1. The algorithm to solve this problem in 7 matches is:

In Round 1, we divide 8 students into 4 pairs, and let each pair have a chess match, creating 4 matches so far (i.e., A vs B, C vs D, E vs F, G vs H).

In Round 2, we have 4 winners from Round 1, and we divide them into 2 pairs and create 2 more matches.

In Round 3, we have 2 winners from Round 2 and only one more match suffices to find the best player.

The minimum number of matches is 4 + 2 + 1 = 7.

It’s not possible to solve this problem in 6 or fewer matches because:

In order to ensure that the highest-rated player is determined, we need to have enough matches to eliminate the rest of the players, which has a number of 7. If there are 6 or fewer matches, we can only eliminate 6 or fewer players, making us unable to identify the highest-rated player.

1. In terms of the ranking of 8 players, there are 8! permutations.

Consider all the permutations are included in a balanced decision tree, where each leaf node represents a permutation.

According to Problem 1(a), a binary tree of height *h* can have at most 2*h* nodes on the last level.

Therefore, the height of this decision tree should be larger than so that all permutations are considered, while the smallest integer bigger than is 16.

It represents that in order to reach any leaf node of this decision tree, there should be at least 16 comparisons, so it’s impossible to solve this problem in 15 or fewer comparisons.

1. The algorithm to solve this problem in 7 matches is:

In Round 1, we divide 8 students into 4 pairs, and let each pair have a chess match, creating 4 matches so far (i.e., A vs B, C vs D, E vs F, G vs H).

In Round 2, we have 4 winners from Round 1, and we divide them into 2 pairs and create 2 more matches. We can also determine 4 losers from Round 1, and similarly we divide them into 2 pairs and create 2 more matches.

In Round 3, we have 2 winners from Round 2 and only one more match suffices to find the best player. We also have 2 losers from Round 2, and it also takes one more match to find the lowest-rated player.

The minimum number of matches is 4 + 2 + 2 + 1 + 1 = 10.

It’s not possible to solve this problem in 9 or fewer matches because:

By using a tournament-style approach, our algorithm in (a) is guaranteed to find the highest-rated player in 7 matches. However, this problem also requires us to find the lowest-rated player. If we adopt a similar approach to (a) and eliminate the lowest-rated player from the beginning, it will require another 7 matches, which adds up to 14 matches in total. However, note that there will be repeated matches in Round 1, because either finding the highest-rated player or finding the lowest-rated player needs 4 matches in Round 1 to eliminate half of them, and it can precisely tell us what the other four players are. Therefore, we can reduce 14 matches to 10 matches in this algorithm. There is no other way to further reduce the number of total matches, because in the following rounds, finding the highest-rated player in a group of 4, or finding the lowest-rated player in another group of 4 (Note that these two groups don’t overlap anymore) both requires at least 3 matches to eliminate the rest of the candidates, so the total number of matches is 10 and it’s impossible to solve the problem using fewer matches.

1. The algorithm to solve this problem in 9 matches is:

First, we follow the algorithm given in (a) to find the best player using 7 matches. Suppose Player 1 is the best player and tournament results are illustrated by the graph below:

A diagram of a network

Description automatically generatedThen, we can find that only Players 2, 3, and 5 lost to Player 1, and all the other players proved to be less strong than them, therefore impossible to be the second-best player. Then, to find the best player among Player 2, 3, and 5, two more matches are needed (Choose any two to have the first match and let the winner compete with the third player).

Therefore, it is guaranteed to solve this problem in 7 + 2 = 9 matches.

It’s not possible to solve this problem in 8 or fewer matches because:

By using a tournament-style approach, our algorithm in (a) is guaranteed to find the highest-rated player in 7 matches. However, this problem also requires us to find the second highest-rated player. If we are using 8 or fewer matches, it only allows us at most one more match to find the second-best player. As discussed previously, there are 3 candidates for the second-best player, so one more match is not enough to eliminate two of them or find the highest rated among them. Two more matches are necessary to guarantee the second highest-rated player, which is why the algorithm above requires 9 total matches, instead of 8 or fewer.