CISC320 - A Day At the Races - Dan Mihovch

This document will contain my final answers and a short description of how I got there including any resources that I used. Also uploaded to canvas will be a PDF to a Notability document where I documented all of my work, thought processes, failed attempts, and eventually, my final solutions.

Notes:

- In my work, I sometimes used 0-based indexing and sometimes used 1-based indexing, though mostly 1-based. Sorry! For my explanations here, I will use 1-based indexing.
- In part 2, if latin square design is not the correct term for what I did, sorry. I simply used the technique I used and then tried to find a name to describe it, and that was the best I got

Part 1:

The most optimal solution that I was able to find for part 1 was 7 races to determine the 3 fastest horses out of 25, while only being able to race 5 at a time. Here was my solution with a bit of my thought process:

- Race 5 disjoint sets of horses, denoted A,B,C,D,E
  - 5 races, 5 total
- Created set W, containing a_1,b_1,c_1,d_1,e_1 (winners of the disjointed races)
- Race set W, producing an ordered set of w_1,...,w_5
  - 1 race, 6 total
  - We now also know that w_1 is the fastest horse
- From here on out, we will use the results of set W's race to transitively eliminate horses
- We can disregard w_4 and w_5, along with their original groups, since they are each faster than their respective group members, and are also themselves slower than w_1,w_2,w_3, prohibiting them from being in the top 3. After this, we are left with 15 horses

- We now can create sets
    - X = {w_1 + w_1's original group}
    - Y = {w_2 + w_2's original group}
    - Z = {w_3 + w_3's original group}
    - Now, $w_1 = x_1$, $w_2 = y_1$, and $w_3 = z_1$
- Since we know that x_1 is the fastest horse, and we only have 2 slots left (2nd + 3rd), we can eliminate x_4 and x_5
- Continuing with that logic, we can also eliminate y_3, y_4, y_5, since if y_1 were to be the second fastest, there would only be 1 slot left (3rd), prohibiting y_3, y_4, y_5 from being in the top 3.
- Again, continuing with the same logic, we can eliminate z_2, z_3, z_4, z_5, since if z_1 were to be the 3rd fastest, there would be no slots left in the top 3, prohibiting them from being in the top 3
- Now, we are left with horses $x_1, x_2, x_3, y_1, y_2, z_1$. We already know x_1 is the fastest horse, so we race the remaining 5, and the top 2 finishers place 2nd and 3rd, respectively
    - 1 race, 7 total
- Resources used:
    - The only resource I used for part 1 of this assignment was ChatGPT, using it **strictly** as a checking tool. I would present a solution, and ask for the model to try and poke holes in it, while not giving me any solutions of its own.

Part 2:

The most optimal solution I was able to find for part 2 was also 7 races to find the median (5th) fastest horse out of 9 while only being able to race 3 at a time. Here is my solution and a bit of my thought process:

- Race 3 disjoint sets of horses, denoted as A,B,C
  - 3 races, 3 total
- Create 3 more sets, X,Y,Z, and populate them using latin square design, such that
  - $X = \{a_1, b_2, c_3\}$
  - $Y = \{a_2, b_3, c_1\}$
  - $Z = \{a_3, b_1, c_2\}$
- Now, we race these 3 sets, yielding ordered sets for each
  - 3 races, 6 total
- We can eliminate the overall winner (index 1) and overall loser (index 3) from each set, since no matter the ordering of each of these races, the result will yield the winner having 5 horses that are slower than it, and the loser having 5 horses that are faster than it, prohibiting either from being the 5th fastest horse.
- We are now left with an unordered set $M = \{x_2, y_2, z_2\}$. We race them, and the 2nd place horse is the 5th fastest overall, and thus we have found our median
  - 1 race, 7 total
- Resources used:
  - Professor Silber gave me the hint of diagonals after 372 lecture, which is what pushed me down the course of using the method I am referring to as latin square design
  - I used the search term in google “finding median from an unordered list”, which consciously led me nowhere, though it did teach me about the quickselect algorithm
  - As with part 1, I used ChatGPT, this time not strictly as a checker, though I did use it for that as well. The only time I used it for something other than checking was to ask (without giving answers), what my professor meant by diagonals, to see if I was missing

something obvious. All it did was tell me that I should use them to eliminate horses (duh).