Hector_Suazo_Writeup

Question 1A

A problem has optimal substructure if it's optimal solution is the sum of it's subproblems. In this question, if we start at N=1 and M=1, there is obviously one solution. While incrementing N or M, more solutions will appear. When uninjured, we increment N by 1 and M remains the same. When injured, we decrement N by 2 and rest for that day, so M will then get incremented by 1. If our N and M both equal 2, we have two possible solutions to reach the max, N=2. We can reach N=2 on the first day, M=1, or the second day, M=2. So we have 3 options for our base case. 1) When N reaches the max N, meaning we reached the max speed we want. 2) When N is 0 or less, meaning the athlete is too injured to run. 3) When M reaches the max M, meaning we reached the end of our time limit.

Question 1C

$$\begin{tikzcd}
 & N_i \ M_k \arrow[dl, "\text{Not Injured}"'] \arrow[dr, "\text{Injured}"] & \\
N_{i+1} \ M_k & & N_{i-2} \ M_{k+1}
\end{tikzcd}$$

N speeds = 6   M days = 2

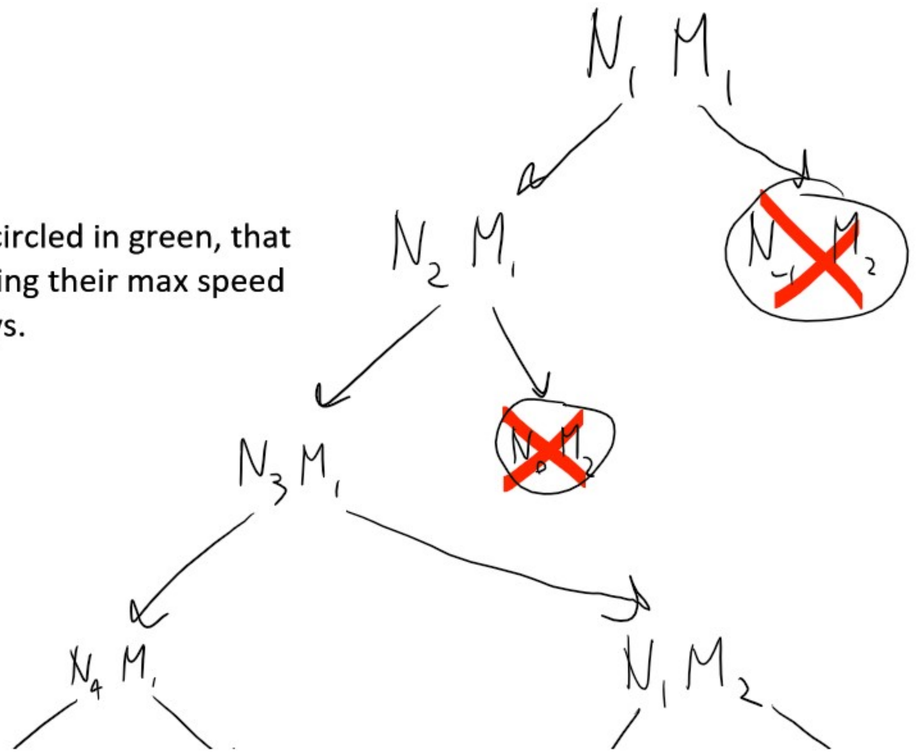
Ends if

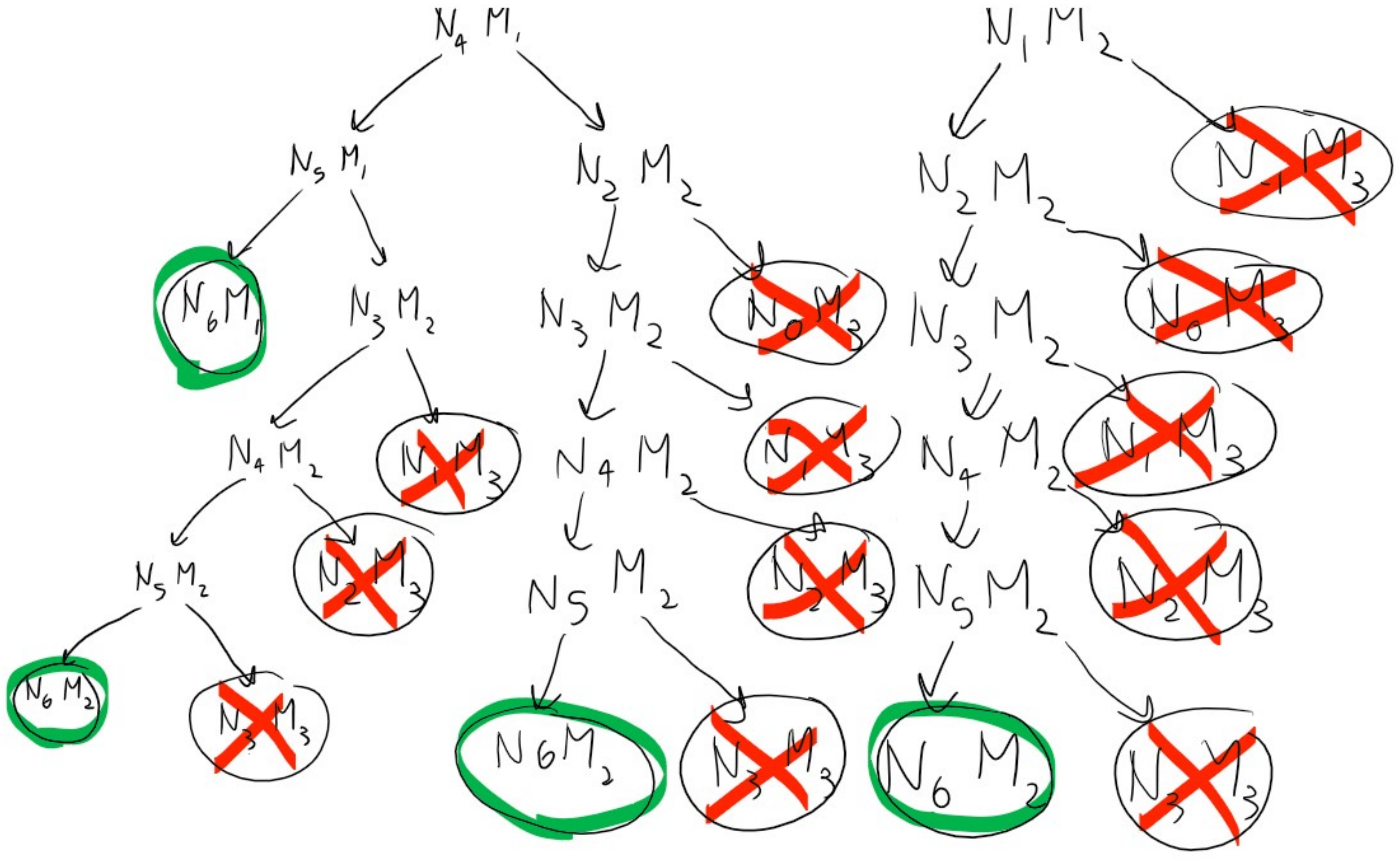
  i <= 0 OR i =6

  OR

  k = 3

There are 4 solutions, circled in green, that show the runner reaching their max speed of 6 in either 1 or 2 days.





Question 1D

<u>There are 15 Distinct Subproblems</u>

N_1M_1
N_2M_1
N_3M_1
N_4M_1
N_5M_1
N_6M_1
N_3M_2 (Repeated 3 times)
N_4M_2 (Repeated 3 times)
N_5M_2 (Repeated 3 times)
N_6M_2
N_2M_2
N_6M_2
N_1M_2
N_2M_2
N_6M_2

Question 1F

We get a solution we solve a subproblem. So when we have a solution we store that solution in an array where the index is the location where we solved that subproblem. This allows us to quickly look through the array if we reached a subproblem that has been already solved.

Question 2A

An optimal substructure comes from the subproblems. In this problem, the subproblems are finding out which cook can do the next step. So the question being asked is, which is the next available cook that can do the most number of consecutive steps.

Question 2B

A greedy algorithm for this problem would be to see who is the next available cook that can do the most consecutive steps and schedule them for those steps. Then repeat until all the steps have been assigned to a cook.