Writeup For Question 2: When Are Teams Eliminated From Playoffs

We solved this problem by a "backward" method. The general idea is that: if team A is top 8 in its conference on the last day of the season, then it must be a "playoffs"; otherwise, we looked 1 day earlier the best scenario/game results for A. If team A could make it to top 8 under the best scenario, then A is eliminated only on the last day; otherwise we repeat this step until we find a date when A could make it to top 8 under the best scenario (among all possible match results after that date). Then the next match date of the date we found is the earliest date when A is eliminated from the playoffs for sure.

We describe in detail how we find the "best scenario" for team A on a certain date and on the last date. The way we calculate the best scenario for A on a certain date is to determine the result of each game on that date in a way that favors A, then rank all the teams in the same conference with A.

The rules to determine the result of a game (team B vs team C):

1. If A is in the game, A wins.
2. If B (or C) is in the same conference with A and C(or B) is not, B(or C) loses.
3. If neither B nor C is in the same conference with A, game result doesn't matter.
4. If both B and C are in the same conference with A, rank among A, B, C needs to be considered.
    1. if rank(B) < rank(C),
        A ranks highest or same as B, B loses.
        A ranks in between, B loses.
        A ranks lowest or same as C, C loses.
    2. if rank(B) = rank(C),
        A ranks higher, whoever wins less wins.
        A ranks lower, whoever wins more wins.
    3. if rank(B) > rank(C), reverse case 1.

With this algorithm, we can calculate the best result on a certain date and with it, we can calculate the best result on the next date until the last day.

Program and Data structure:
We use python to solve this problem.
We convert the original data in the following data structure

(1) dates: a list of each date (string). In total 162 days in 2016-2017 season.
(2) matches: a list of dictionaries. In each dict, keys are tuple of 2 teams and value is 0 when 1st team wins and 1 when 2nd team wins.

(3) daily_scores_east, daily_scores_west: a dict. Keys are teams and values are list of tuple (num of wins, num of losses).
(4) daily_rank_board_east, daily_rank_board_west:a list of dict with length 162, in each dict, keys are teams and values are ranks.

Functions:

EliminationDate(team,dates,matches,daily_scores_east,daily_scores_west,daily_rank_board_east,daily_rank_board_west):separate teams into each conference.

Find_EDate(team,dates,matches,daily_scores,daily_rank_board):by calling Best_Final_Rank until the best final rank of a certain team is better than 8, return the date.
Best_Final_Rank(team,date,dates,matches,daily_scores,daily_rank_board): by calling Best_daily_Rank until the last date, return the best final rank.

Best_daily_Rank(team,teams,Best_Rank,Best_score,daily_match): given the current rank (Best_rank) and scores (Best_score) of a certain conference, and the match schedule on a certain date, return the best possible rank for a team on a certain date.

Rank(Best_score): calculate rank with score of each team.