CS 4820, Spring 2019	**Homework 2, Problem 2**
Name: Kevin Klaben NetID: kek228 Collaborators:	

(2) *(10 points)* In the sport of American football, teams move a ball toward the opposing team's goal line in a sequence of plays called an offensive drive. Subject to some simplifications, the game has the following rules.

1. An offensive drive starts n yards away from the other team's goal line. In American football, n is often but not always equal to 80.
2. To retain possession of the ball, the team must move forward at least k yards in its first d plays, called "downs". In American football, $k = 10$ and $d = 4$.
3. More generally, certain plays in an offensive drive are called "first downs". The initial play in the drive is a first down. A subsequent play is called a first down if and only if the total yardage accumulated since the preceding first down is greater than or equal to k.
4. There are three ways that a drive could end.
   - If the total yardage accumulated in the drive is greater than or equal to n, the team scores a touchdown.
   - If the team has completed d plays since the last first down, and the total yardage accumulated in those d plays is less than k, the team loses possession.
   - Any play could result in a "turnover". If this happens, the team loses possession immediately.
5. On any given play of the drive, the team needs to decide whether to try a passing play or a running play. The probability of a turnover, and the distribution of the random number of yards gained in the event that the play is not a turnover, depend on whether the team chooses passing or running. We will assume that the number of yards gained is always a non-negative integer.[1]

A *football strategy* is a strategy for choosing between a running or passing play in every possible situation that may arise during an offensive drive. A strategy is *optimal* if it maximizes the probability of scoring a touchdown.

For an integer y in the range $0 \leq y \leq n$, let p_y denote the probability of gaining y yards on a passing play, and let r_y denote the probability of gaining y yards on a running play. Let p_{n+1} and r_{n+1} denote the probability of a turnover on a passing or running play, respectively. Assume these are non-negative rational numbers and that $\sum_{y=0}^{n+1} p_y = \sum_{y=0}^{n+1} r_y = 1$. Design an algorithm which is given the parameters n, k, d and the probabilities p_y, r_y for $y = 0, 1, \ldots, n+1$, and which determines whether the optimal football strategy chooses to run or to pass at the start of an offensive drive, n yards away from the opponent's goal line. Your algorithm should work for *any* values of n, k, d, not only for $n = 80, k = 10, d = 4$.

Football(n,k,d)

[1] In the actual sport of football, it's possible for a play to gain a negative number of yards. We are deliberately ignoring this possibility, for the sake of simplicity.

```
norig=n korig=k dorig=d
tp=float arr[n+1][k+1][d+1] filled with null
tr=float arr[n+1][k+1][d+1] filled with null
RECFOOT(n,k,d)
if tp[n][k][d]¿tr[n][k][d] then return pass
else
return run
endFOOTBALL
fun RECFOOT(n,k,d)
if tr[n][k][d]!=NULL and tp[n][k][d]!=NULL then
   return max(tr[n][k][d], tp[n][k][d])
   if n<=0 then
      return 1
   else if d=0 then
      return 0
   end if
   prun=0 ppass=0
   for i in range(0,n do
         (m,l)=RECGERRY(i+j+1,k-1)
         if i>=k then
         ptd=RECFOOT(n-i,korig,dorig)

      else
         ptd=RECFOOT(n-i,k-i,d-1)
      end if
      prun=prun+r_i*ptd
      prun=prun+p_i*ptd
   end for
   tr[n][k][d]=prun
   tp[n][k][d]=ppass
end if
```

Runtime Analysis: This algorithm memoizes two 3 dimensional arrays to represent the touchdown probabilities. The memoization happens throughout the running of the algorithm each time that a recursive call is made for which no output exists. As each memoization of a new value occurs, it involves potentially iterating over a maximum of n elements. Thus the overall runtime for the memoization is O(nkd*n)=O(n^2kd). As the remainder of the function is simply constant time operations, the overall runtime of the algorithm is O(n^2kd).
Proof of Correctness: Claim: the algorithm produces the correct choice of football strategy in all situations. Proof: For the sake of contradiction, supposed the algorithm chose the less likely play to produce a touchdown. Thus, the probability of scoring a touchdown in the optimal choice is higher than the touchdown probability in the chosen play. As these probabilities are stored in the memoized 3d array, and compared, by showing that the values of the memoized array are correct is sufficient to show the algorithm is correct.
We can prove that the memoized array is correct by induction on n (number of yards remaining to gain overall. IH: Tr and Tp contain the highest probability of scoring a touchdown.

Base Case: given that n is zero, show that tr and tp will contain the highest probability of scoring a touchdown. Proof: Well as n is the yards remaining to scoring a touchdown, the probability that is always returned is 1 as when n=0 a touchdown has already been scored. Thus all spots iin the 3d arrays tr and tp will contain 1 when n is zero which is the probability of scoring a touchdown in that situation.
Inductive case: We want to show that for the entries in the nth row are correct given that the rows for smaller n's already contain the highest probabilities possible in such situations. Proof: At n the algorithm calls for iterating over the from 0 to n yards gained on a play. We can easily access the probabilities of gaining each number of yards on that specific play by running or by passing through the Probabilities in P, but we must consider the likelihood of scoring a touchdown following this play. As this requires a recursive call to a smaller or equal value of n (in which case the k and d get smaller and thus progress is made), we can return the optimal probabilities as we assumed that all entries for smaller n's have already been computed. As the both pieces are present to compute the sums over all possibilities we can fin the optimal probability and add it to the table in the spot n,k,d and thus the memoized tables will contain the correct values and the overall algorithm will output the correct play choice in all situations.