Winning at Sports

Mark Leyfman

Problem Statement

- In the game of *Sports*, the object is have more points than the other team after a certain amount of time has elapsed.
- You're very good at *Sports*, and consequently you always win. However, you don't always achieve victory the same way every time.

Problem Statement (cont.)

- In a **stress-free** victory, you score the first point and from then on you always have more points than your opponent.
- In a **stressful** victory, you never have more points than your opponent until after their score is equal to their final score.

Problem Statement (cont.)

- Given the final score of a game of *Sports*, how many ways could you arrange the order in which the points are scored such that you secure a **stress-free** or **stressful** win?

IO

- You are given T games' final scores
- Output the number of ways you can achieve a **stress-free and stressful** win modulo $10^9 + 7$
- $T \leq 100$, Scores limited to 2000, 6 minutes limit

Examples

- If the final score is 3-1, there are 2 ways to achieve a stress-free victory: WWWL, WWLW, and 1 way to achieve a stressful victory: LWWW, so output is 2, 1
- If the final score is 3-2, there are 2 ways of achieving a stress-free victory: WWWLL, WWLWL, and 2 ways of achieving a stressful victory: LLWWW, LWLWW, so output is 2, 2

Questions?

Facebook's Solution for Stress-Free Wins

Let f(u, t, U, T) = # of ways to achieve stress-free victory when we have u points, opponent has t points, and final score is U-T.

We then calculate f(0, 0, U, T) as follows:

f(U, T, U, T) = 1 (done)

f(u, T, U, T) = 1 if u > T, 0 otherwise

f(U, t, U, T) = 1

f(u, t, U, T) = 0 if u > 0 and u <= t (not stress-free)

f(u, t, U, T) = f(u+1, t, U, T) = f(u, t+1, U, T)

Facebook's Solution (cont.)

Proof of correctness:

f(U, T, U, T) = # of ways to achieve a stress-free victory when you have U points, the opponent has T points, and final score is U-T is 1 because there are no more points to score.

f(U, t, U, T) = # of ways to achieve a stress-free victory when only opponent can still score: 1, since you can't score any more points

Facebook's Solution (cont.)

f(u, T, U, T) = 1 if u > T, as opponent is done scoring, and you still have some left to score. If u <= T, then the win is not stress-free and so produces 0 ways.

f(u, t, U, T) = 0 if u > 0 and u <= t, since opponent has more points, meaning that this isn't a stress-free victory.

Facebook's Solution (cont.)

In all other cases, either you or your opponent might be able to score, so you can add the two possibilities together:

$$f(u, t, U, T) = f(u+1, t, U, T) + f(u, t+1, U, T)$$

Since each step increases u, t and U, T are finite upper bounds, the algorithm terminates when it reaches those bounds, thus producing the correct result.

Facebook's Solution for Stressful Wins

Let g(u, t, U, T) = # of ways to achieve stressful victory when we have u points, opponent has t points, and final score is U-T.

We then calculate g(0, 0, U, T) as follows:

g(U, T, U, T) = 1 (done)

g(u, T, U, T) = 1

g(U, t, U, T) = 0 (not stressful)

g(u, t, U, T) = 0 if $u > t$ (not stressful)

g(u, t, U, T) = g(u+1, t, U, T) = g(u, t+1, U, T)

Facebook's Solution for Stressful Wins

Proof of correctness:

Basically, same as the first one.

Analysis of Facebook's Solution

DP takes U * T memory and U * T time for each query, resulting in roughly:

2*2000*2000*100 = 800m operations

Questions?

Stressful Victories

- If the score was 2-1, there is 1 way to win: LWW
- If the score was 3-1, there is 1 way to win: LWWW
- If the score was N-1, there is still 1 way to win: LW…W

Stressful Victories (cont.)

- If the score was 3-2, there is 2 ways to win: LLWWW, LWLWW
- If the score was 4-2, there is 2 ways to win: LLWWWW, LWLWWW
- If the score was N-2, there is still 2 ways to win: LLW…W, LWLW...W

Stressful Victories (cont.)

Conclusion: winner's score doesn't matter, as there is always excess W's. Therefore, our answer should only depend on the loser's score.

Loser's score	Resulting Ways
0	1
1	1
2	2
3	5
4	14

Catalan Numbers

Catalan Numbers

The sequence 1, 1, 2, 5, 14, 42, 132, 429, … is known as the Catalan number sequence, and commonly shows up in counting recursively defined objects.

Applications of Catalan Numbers

- Number of expressions with n correctly matched pairs of “(” and “)”
- Number of full binary trees with n+1 leaves
- Number of monotonic paths along grid which do not pass the diagonal
- Number of ways a convex polygon with n+2 sides can be cut into triangles with straight line cuts

Applications of Catalan Numbers

- Number of ways a losing team can cause the winning team to win stressfully
  (Our problem!)
- There are many, many more (70+) interpretations available on OEIS (On-Line Encyclopedia of Integer Sequences) and Wikipedia

Formulas for Catalan Numbers

$$C_n = \frac{(2n)!}{(n+1)!n!} = \frac{2(2n-1)}{n+1} C_{n-1} = \sum_{i=0}^{n-1} C_i C_{n-1-i}$$

Stressful Wins Conclusion

- We can calculate all 2000 possible values by using the middle formula and storing the intermediate results in O(n) time

Questions?

Stress-free Wins

- We can look for patterns in the win numbers by making a triangle
- Define F(a, b) = # of ways team can win if final score is a-b (winning team has a points, losing team has b points)

Stress-free Wins (cont.)

F(1, 0) = 1					
F(2, 0) = 1	F(2, 1) = 1				
F(3, 0) = 1	F(3, 1) = 2	F(3, 2) = 2			
F(4, 0) = 1	F(4, 1) = 3	F(4, 2) = 5	F(4, 3) = 5		
F(5, 0) = 1	F(5, 1) = 4	F(5, 2) = 9	F(5, 3) = 14	F(5, 4) = 14	
F(6, 0) = 1	F(6, 1) = 5	F(6, 2) = 14	F(6, 3) = 28	F(6, 4) = 42	F(6, 5) = 42

Stress-free Wins (cont.)

F(1, 0) = 1					
F(2, 0) = 1	F(2, 1) = 1				
F(3, 0) = 1	F(3, 1) = 2	F(3, 2) = 2			
F(4, 0) = 1	F(4, 1) = 3	F(4, 2) = 5	F(4, 3) = 5		
F(5, 0) = 1	F(5, 1) = 4	F(5, 2) = 9	F(5, 3) = 14	F(5, 4) = 14	
F(6, 0) = 1	F(6, 1) = 5	F(6, 2) = 14	F(6, 3) = 28	F(6, 4) = 42	F(6, 5) = 42

Stress-free Wins (cont.)

We notice that there is a recurrence:

$$F(a,b) = \sum_{i=0}^{b} F(a-1,i)$$

Therefore, we can make a 2000x2000 array, calculate F(2000, 1999) with memoization, and then each query takes O(1) time after that.

Stress-free Wins (cont.)

However, constructing this table takes O(mn) time, and since the number of query's isn't very large, it is somewhat slow.

Can we do better?

But First, Questions?

Bertrand’s Ballot Theorem

Bertrand’s Ballot Theorem

In an election where candidate A receives p votes and candidate B receives q votes with p > q, what is the probability A will be strictly ahead of B throughout the count?

Answer: $$\frac{p-q}{p+q}$$

Bertrand's Ballot Theorem (cont.)

Since total number of voting orders is:

$$\binom{p+q}{q}$$

Then the total number of ways to win in this particular way is:

$$\binom{p+q}{q}\frac{p-q}{p+q}$$

Stress-free Winning Summary

We can calculate an individual value in O(m) time, where m is the losers score, with the following formula for a binomial coefficient:

$$\binom{n}{m} = \frac{n(n-1)\ldots(n-(m-1))}{m(m-1)\ldots1} = \prod_{i=1}^{m} \frac{n+1-i}{i}$$

Winning at Sports Summary

- We can use Bertrand's Ballot Theorem to calculate the number of ways to achieve stress-free wins in linear time
- We can use Catalan Numbers to calculate the number of ways to achieve stress-full wins in linear time
- Overall problem complexity: linear

That's all folks!
Questions?