

Programming Assignment 1

210050155 - Tanay Vineet Tayal

March 2023

Algorithm

I have created 2 arrays of pairs of integers, one contains optimal values of A and the corresponding optimal value of B while the other contains optimal value of A=B and the corresponding optimal value of A. It's like, $besta[i][j]$.first stores the optimal value of A if player 1 is at i th vertex and there are j moves left while $besta[i][j]$.second stores the optimal value of B if player 1 was at the i th vertex and made an optimal move, then the optimal value of B i.e. player 2. Similarly for $bestb$.

The optimal value is calculated based on the maximum possible difference in score at a particular place.

$$(1) besta[i][j].fi = \max_{\{k \neq i\}} w_j k + bestb[k][j].se - bestb[k][j].fi$$

$$(2) besta[i][j].se = bestb[k1][j].fi$$

$$(3) bestb[i][j].fi = \max_{\{k \neq i\}} w_j k + besta[k][j-1].se - besta[k][j-1].fi$$

$$(4) bestb[i][j].se = besta[k3][j-1].fi$$

Where $k1 = \text{argmax}$ of the first statement and $k3 = \text{argmax}$ of the third statement

We now have created a bottom up approach for filling our DP table i.e. the arrays. $besta[i][0]$ and $bestb[j][0]$ for all i, j are 0,0. There are a total of $O(n \times m)$ entries in the table and it takes $O(n)$ time to compute 1. Thus, overall time complexity is $O(n^2 \times m)$