Partial Solution using dynamic programming

A state of the game is fully described by the position of 007, the position of Dr. Null, and whose turn it is. Using dynamic programming you can calculate for each of these $2n^2$ states (starting from the end-position) who would win, provided that both play optimally. This solution takes $\mathcal{O}(n^2 + nm)$ time and will give you 30 points.

Full Solution

First, determine – using breadth-first-search – the distances a_1, a_2 from 007's start point to the servers as well as the distances b_1,b_2 from Dr. Null's starting position to the servers. This takes $\mathcal{O}(n+m)$ time. Furthermore, let $w_1 = b_1 - a_1$ and $w_2 = b_2 - a_2$. These are the maximum times 007 could wait, if there was only one server. With two servers, there are two cases:

- If $w_1 = w_2 - 1$, the optimal strategy for 007 is to wait w_1 turns and then go straight to server 1. If $w_1 < 0$, then 007 cannot win at all. The same argument holds if $w_1 = w_2 + 1$.
- If $w_1 = w_2$, things get more tricky. Note that the maximum waiting time is always either w or $w - 1$ (if this is negative, 007 can not win). Let c (and d) be the maximum number of steps 007 (and Dr. Null) can make while shortening her/his distance to both servers at each step. If $c + w \geq d$, the answer is w, if $c + w < d$, the answer is $w - 1$. The numbers c and d can again be determined in $\mathcal{O}(n+m)$ time.

Solving only the first case and always outputting $w - 1$ in the second case will give you 30 points. Combining this with the DP solution will give you up to 51 points.