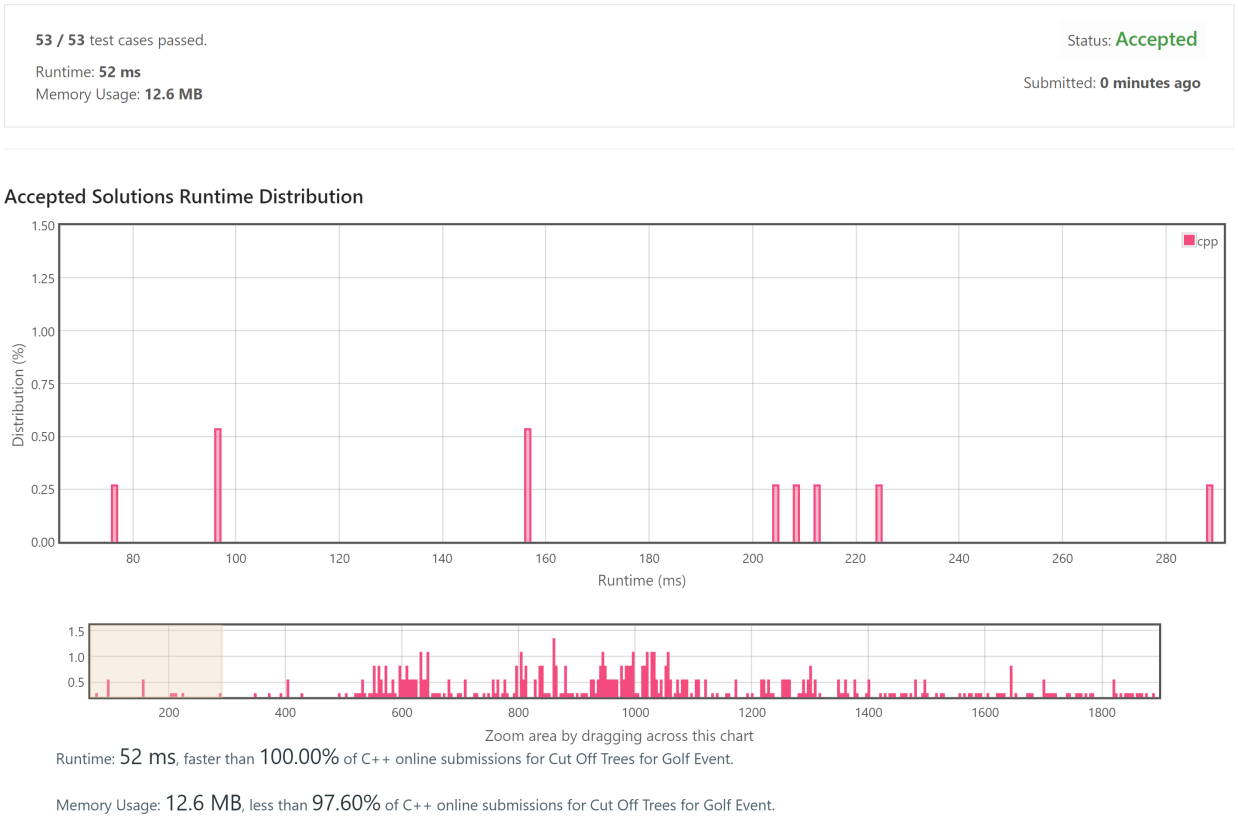
for simplicity assume $n = m$.
1. repeatedly use bfs to find the shortest path. we need to compute $O(n^2)$ shortest paths, and each takes $O(n^2)$ time. $O(n^4)$.
remark. hadlock's algorithm is faster in practice.
2. use divide and conquer. divide the grid into two parts using the middle line. for each point on the middle line, compute the SSSP from it to each point in the grid. for each query, if the source and the target are on different sides, then the shortest path must intersect the middle line. otherwise the shortest path may or may not intersect the middle line, and we need to recurse on both sides. the running time is $T(n) = 4T(\frac{n}{2}) + O(n^3)$, which is $T(n) = O(n^3)$.
This problem is similar to [ZJOI2016]旅行者.




remark. fully dynamic shortest path on grid, $\tilde{O}(n)$ time per query(?)

# References