HKOI 2015/16 TFT Q2 Solution

Problem Setter : Alex Tung

Comments

- Standard problem in graph theory
- Expected “InstanceKill” by experieced candidates
- Expected more full-scorers

Idea of solution

- In order to maximize the path value, we can...
- Binary search on the path value (fix the path value) and see is that possible

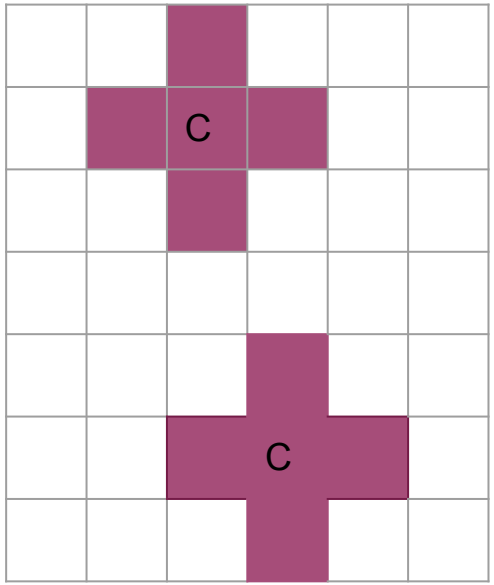
Idea of solution

e.g. Let path value = 2, we cannot visit the cell with cell value <= 1

Cells with “C” denote there is a crow.
Cells in purple denote their cell value <= 1

Check whether Alice can visit
the bottom-right corner from the top-left corner
if she cannot visit the cells in purple

If she can visit the destination, increase the path value
If she cannot, reduce the path value



Idea of Solution

- Your mark depends on the time complexity of this algorithm
- Different implementation results in different time complexity

Solution 1

- Find the cell value of each cells by direct looping ------- O(NMK)

  cellvalue[i][j] = min(dist_between(i, j, crowX[l], crowY[l])) for 1 <= l <= K

- Binary search on answer + BFS on the grid from (1, 1) to (N, M) ------- O(NM lg K)
- Time complexity O(NMK + NM lg N) = O(NMK)
- Passes subtask 2 only

Solution 2

- Find the cell value of each cells by multi-sources BFS ------ O(NM)

  Push every crows in a queue at first
  Find the shortest path between each cells to any crows
  Shortest path = cellvalue

- The else is same as solution 1

- Time complexity : O(NM + NM lg N) = O(NM lg N)

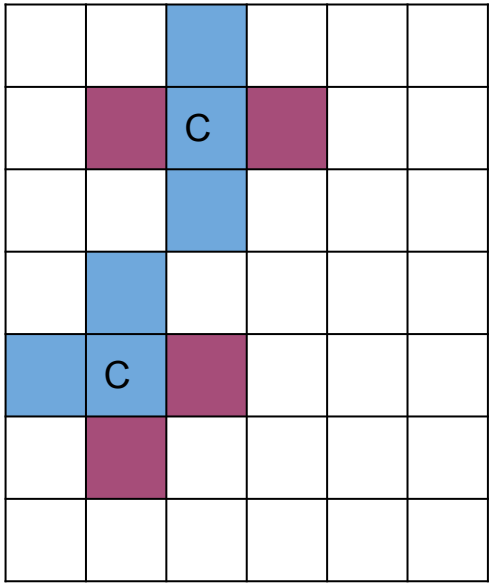
- Passes subtask 3 (and subtask 4 with some constant optimization)

Intended Solution

- To check can Alice visit (N, M) from (1, 1)
- Instead of BFS from (1, 1) to (N, M)
- We can find if there any “block” in the grid

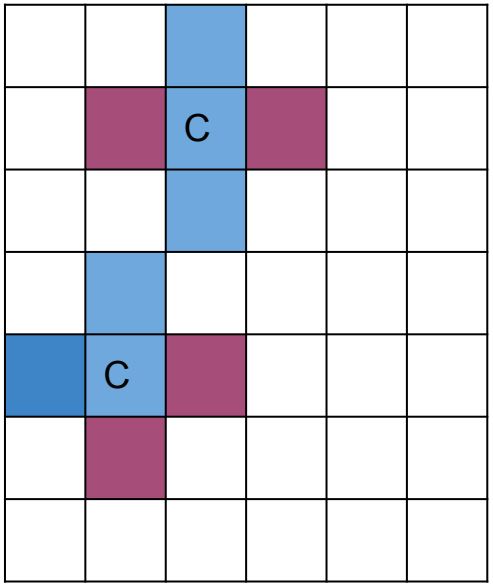
- There is a block in the grid means that the purple cell divided the grid into 2 or more inconnect regions

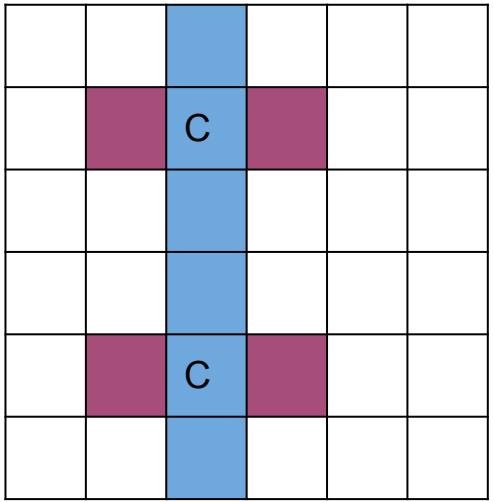
- Which mean, if we visit the purple cells only (Assume two purple cells is connect if they share a same corner) we can go from one boundary to another boundary of the grid if and only if there is a block

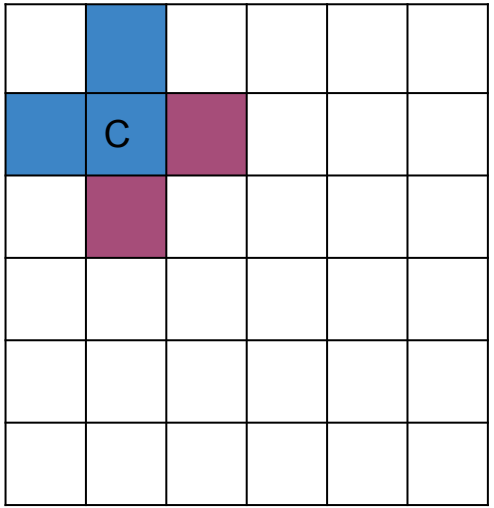


Intended Solution

Grids with a block





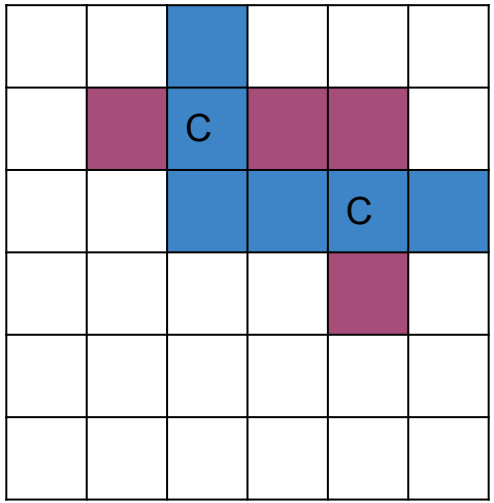


Intended Solution

Be careful, we don’t consider the following as a block because (1, 1) (N, M) still connect

The grid has a block if and only if :

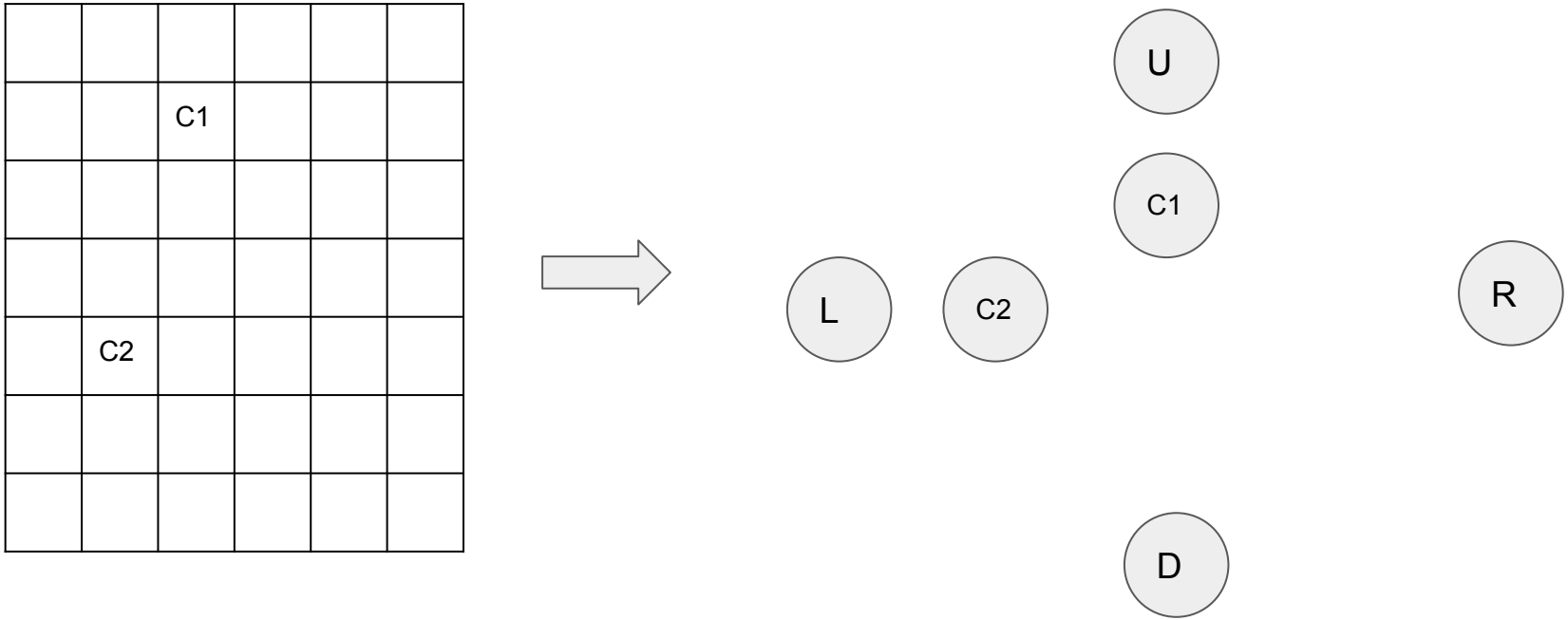
1. We can go to the left, bottom from the top boundary
2. We can go to the left, bottom from the right boundary



Intended Solution

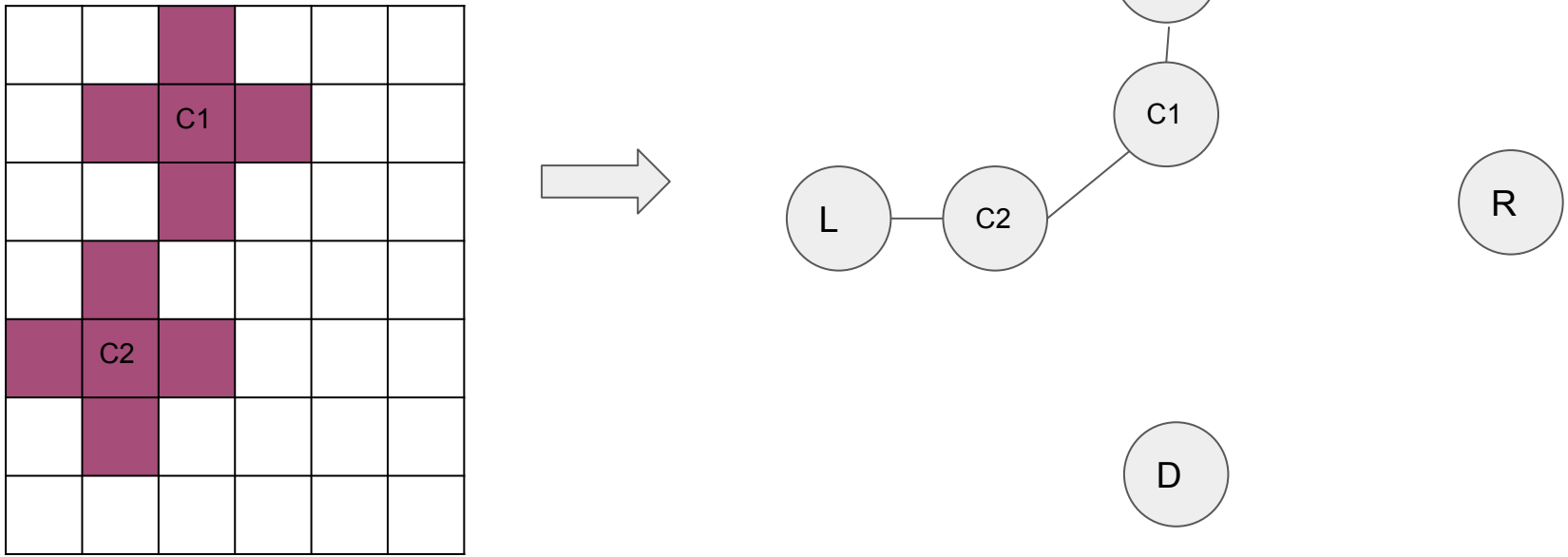
- How to check if the grid has a block?
- We consider the crows and the boundary as nodes to build a new graph

Intended Solution



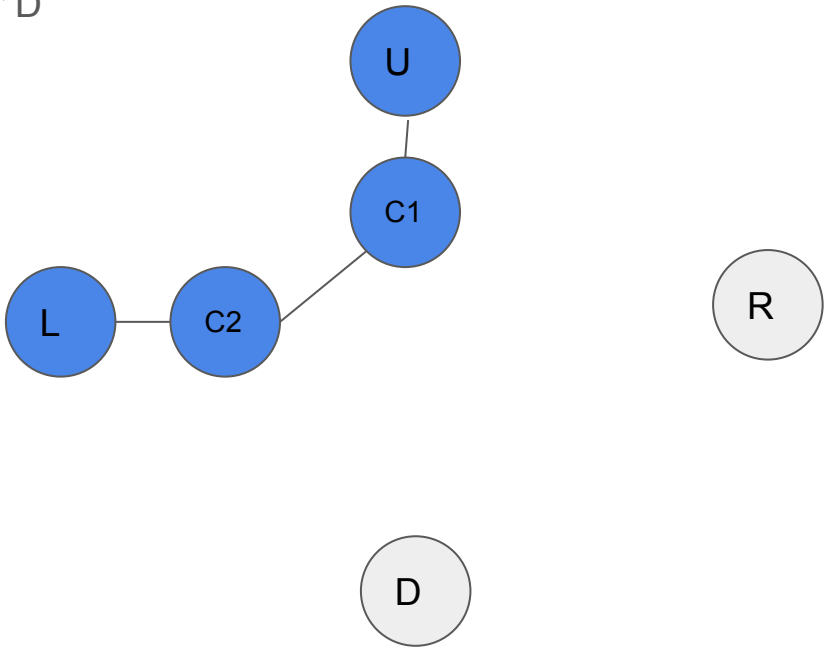
Intended Solution

- We add edge between 2 node if
  the purple area of them share at least a same corner



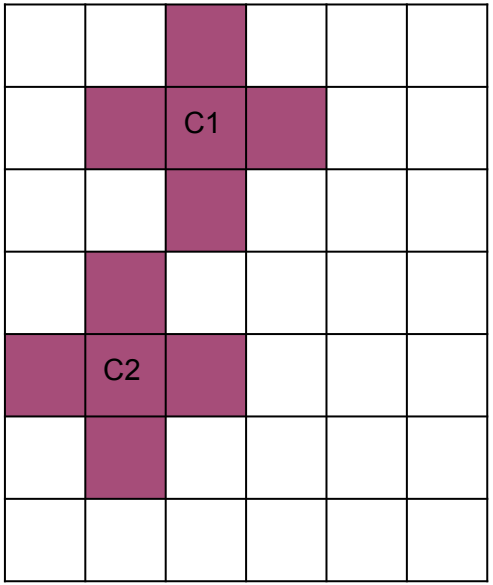
Intended Solution

Then we can BFS from U, find if it connects to L or D
and also BFS from R, find if it connects ot L or D



Intended Solution

- Two shapes share at least a same corner if……
- |row[i] - row[j]| + |col[i] - col[j]| <= mid * 2 (generally)
- need special handle if they have same row[] or col[]



Intended Solution

Number of nodes in this graph = 4 (boundary) + number of crows
= O(K)

Number of edges in this graph = O(K*K)

Therefore, time complexity of building and BFS on this graph is O(K*K)

Combining this with the binary search, the time complexity is O(K * K lg N)

Expected Score : 100 :)