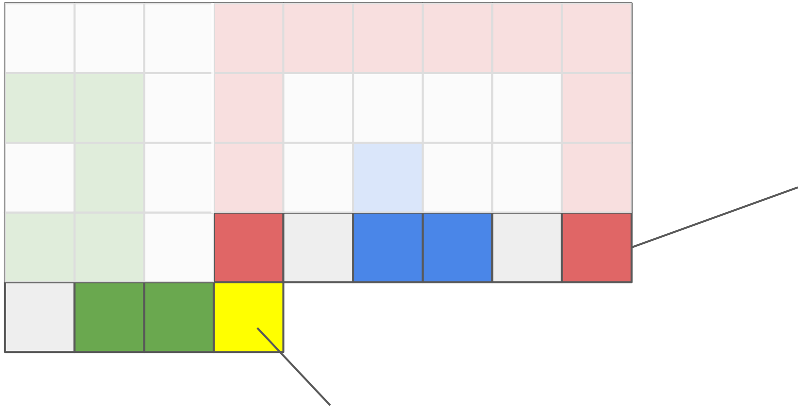
Window Shopping

- Use dynamic programming to decide for each cell (i, j), in the row-major order, if it is selected as a shop or it should be left empty as a hallway.
- The DP state counts the maximum number of windows, and keeps track of:
  - The connected components (CC) of hallways. The state records for each column which CC the lowest cell in that column belongs to.
  - Which components have connected to the escalators U/D. We can track two special CCs for U and D respectively, i.e. CC(U) and CC(D).
  - Eventually, we should have exactly one component that is equal to CC(U) as well as CC(D).
- If we disallow a CC to be disconnected from U or D, we can safely assume that:
  - Any cells left empty are hallways. Therefore any edge between a hallway and a shop has a window.
  - This is equivalent to filling the disconnected CC with “shops”, without installing any windows.

State transition



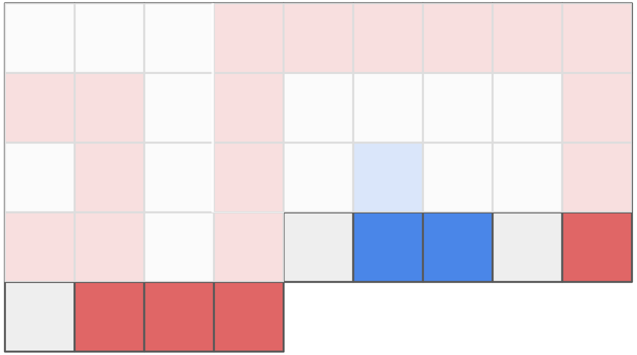
Current Cell

The non-transparent cells represent the DP state that track the CCs.

There are 3 hallway CCs: red, blue, green.

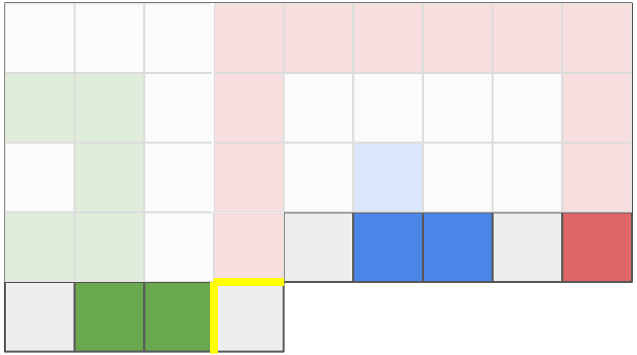
The grey cells are either pillars or shops.

Choice 1: Make a hallway



Merge the two adjacent red and green CCs into one CC.

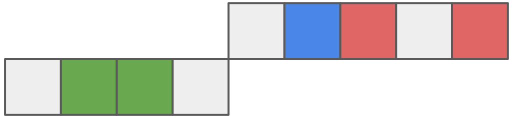
Choice 2: Make a shop



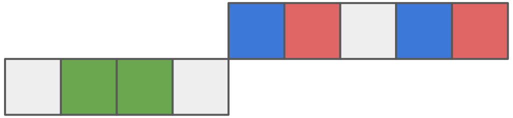
The shop prevents the red and green CCs from connecting.

The shop creates two windows, one to the top and one to the left.

- The total number of states is about 3.3 * 10^5
  - The number of columns <= 9. If C > R, we can rotate the floor by 90 degree so that C <= 9.
  - The state includes information about CC(U) and CC(D), and its count is thus bigger than the typical number of DP states that track CCs.
- We can trim invalid states:
  - (a) Two adjacent cells cannot belong to different CCs in a DP state. This is required, otherwise the number of states is much bigger.
  - (b) Two CCs cannot interleave. This trims about 2 * 10^4 states and is optional.
- Total time: O(RC*S*U)
  - S is the number of states.
  - U is the cost to maintain and update the CCs in a state, which is typically O(C).
  - This is roughly 99 * 3.3 * 10^5 * 9 ~= 2.94 * 10^8.



(a)



(b)