

Project 2B: CSP Tile Placement

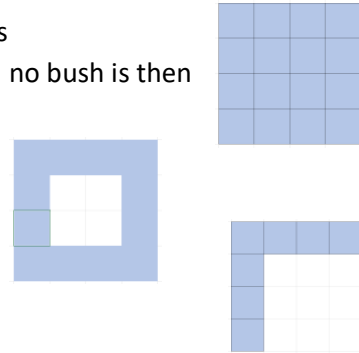


Also see the “GeneralInstructions.pdf”

Note: Individual Project

Given

- You are given a landscape on which certain “bushes” grow, marked by colors: 1, 2, 3, 4.
- The landscape is of square shape, so, it might be 100 x 100 or 200 x 200 etc.
- You are given a set of “tiles” which are of three different shapes. The tiles are 4 x 4. One tile only covers part of the landscape. Here are the shapes
 - Full Block: A “full block” tile covers the full 4 x 4 area, and no bush is then visible in that patch.
 - An “outer boundary” tile covers the outer boundary of the 4 x 4 area, and any bush in the middle part is visible.
 - An “EL” shaped tile covers only two sides of the area.
- You are given a “target” of which bushes should be visible after you have finished placing the tiles.



Observations

- The total tiles cover the entire landscape. However, depending on which tiles are placed where, different parts of the landscape, and hence different bushes are visible.
- The number of tiles equals the size of the area divided by the size of tile. So, for 20 x 20 landscape, you are given 25 tiles.

Input Files

Structure of the input file is as follows.

- Landscape is given in a space delimited, new line separated.
- Tiles in terms of counts by different shapes.
- Target of how many different bushes should be visible.

Many input files can be found at

<https://github.com/amrinderarora/ai/tree/master/src/main/resources/csp/tileplacement>

More input files can be generated using

<https://github.com/amrinderarora/ai/blob/master/src/main/java/edu/gwu/cs/ai/csp/tileplacement/TilePlacementProblemGenerator.java>

Algorithm

Write a CSP algorithm to solve this problem. The CSP algorithm should have the following components:

- Search algorithm to solve the CSP
- Heuristics (min remaining values, least constraining value)
- Constraint propagation using AC3.