Hanooy Maps

a $\varnothing hr$ test task.
v 0.5.1

Ilona Prikule
<ilona@doma.dev>

Jons Mostovojs
<jonn@doma.dev>

May 2023

1 Introduction

Thank you for agreeing to complete this $\varnothing hr$ task! It is designed to take around an hour to complete for a proficient developer. Unless otherwise specified, we suggest spending ≈ 15 minutes to read this document and come up with the solution idea, ≈ 30 minutes to implement the solution, and ≈ 15 minutes to test it.

You are given a project which has some functions which are yet not implemented. You have to fill in implementations of these functions following the specification given in this document.

Note that there may be bugs in the skeleton implementation. If you find an inconsistency, consider the specification correct and the implementation faulty.

2 The gist

Hanooy Inc. is making colourful tower theme parks! Each theme park is given a limited rectangular space. Your task is to plan a top-down map of the theme park, given a set of towers and the colourings of their layers.

The more towers you connect together in clusters, the more awe-inspiring your theme park will be and the more tourists will visit it!

The bases of towers are all of the same size, even though they may have different amounts of levels, each of which is painted in some colour. Consequetive layers are always coloured differently, but the same tower may have two layers that share a colour.

We hope that you'll find the most awe-inspiring tower placement. Good luck!

3 Neighbours

Towers K and L are neighbours if and only if there is at least one pixel $p \in K$ and $q \in L$ such that p and q share one of the coordinates and the other coordinate of these pixels differs exactly by 1. In other words: at least one pixel in K shares an edge with at least one pixel in L.

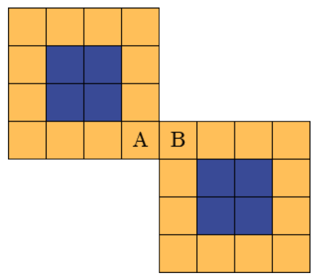

Figure 1: *Pixel marked "A" has shares an edge with pixel marked "B", thus these two towers are neighbours.*

3.1 Valid Placement of Neighbouring Towers

3.1.1 Primality of Towers

If towers K and L are neighbours, and the topmost leftmost pixel of K lies to the left of the topmost leftmost pixel of L, then K is primary and L is secondary, unless the topmost leftmost pixel of L lies to the top of the topmost leftmost pixel of K. In that case, L is the primary and K is the secondary. In other words, primality goes from top to bottom and from left to right.

3.1.2 Validity of Placement

If a primary tower A with layers $A^l_{1\ldots n}$ is a neighbour of a secondary tower B with layers $B^l_{1\ldots m}$, then there exists a non-empty sequence of layers $\alpha = l_1, \cdots, l_i$ such that both the outermost layers of A $(A^l_1, \cdots, A^l_i)$ and the innermost layers of B $(B^l_{m-l+1}, \cdots B^l_m)$ are both precisely α.

Towers can't overlap.

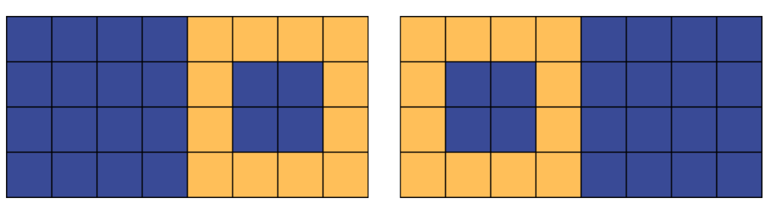

Figure 2: *Left placement of two 4x4 towers is valid, whereas the right one isn't.*

4 Inputs

An input contains a list square 24-bit bitmaps representing available towers and two integers c_x, c_y, representing the canvas size into which you must fit some or all of towers.

Example `input.json`:

```
{
    "ziggurats": ["./mono.bmp", "./di.bmp"],
    "c_x": 8,
    "c_y": 4
}
```

(a) mono.bmp (b) di.bmp

Figure 3: *Files in the same directory with* `input.json`

5 Outputs

Outputs is a list of bitmap indexes together with x, y coordinates of each bitmap. If you couldn't fit a bitmap into the canvas, don't include it into this set.

As in other computer applications, $(0, 0)$ is topmost leftmost pixel! Thus, when we say that point $a = (x_a, y_a)$ lies to the top of point $b = (x_b, y_b)$, we mean that $y_a < y_b$. When we say that it lies to the left of point b, we mean that $x_a < x_b$.

Example output, which your submission should print into STDOUT:

```
[
    {"id": 0, "x": 0, "y": 0},
    {"id": 1, "x": 4, "y": 0},
]
```

6 Scoring

It is said that if a tower is a neighbour of another tower, then they belong to the same cluster.

Each tower X contributes the K^g points to the total score, where K is the size of of the cluster it belongs to, and g is the amount of neighbours of X.

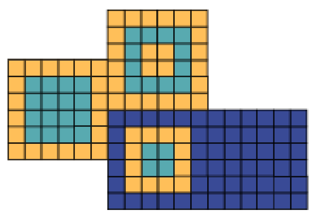

Figure 4: *4 towers form a cluster, 1 of those has 3 neighbours, 2 of those have 2 neighbours and 1 of those has 1 neighbour. Thus, the total score for this cluster is* $100 = 4^3 + 2 \times 4^2 + 4$.

7 Submissions

Your submission will be competing against a bunch of naive solvers (implemented by us) and against solvers, submitted by other candidates. To qualify for a middle-level position, you're expected to beat all the naive solvers. To qualify for a junior position, you're expected to produce a working submission that scores comparable to an average naive solver.

7.1 Running the solvers

The solvers are ran once on each of the different test configurations inside Docker. Each invocation shall be given approximately, but not less than, 1 second of wall time to print out solution into the STDOUT. Along with your submission running, exactly one of the CPU cores shall be allocated to running a naive solver. Aside from that, you can expect an overall idle server with at least three cores available for your solver.