

Tiling the bathroom

Combinatorial Optimization course, FEE CTU in Prague. Created by [Industrial Informatics Department](#).

Motivation

You accidentally dropped a bowling ball in your bathroom, which cracked a significant part of the floor. You successfully removed the debris, but now you must fill the resulting rectangular hole with new tiles. However, the tiles are pretty costly, so you asked the neighbors whether they have any spares. Finally, you acquired several different rectangular tiles whose area is equal to the area of the hole. Are you able to repair the floor?

Input

You are given the following:

- Rectangle hole in your bathroom of dimensions A_x, A_y
- n different square tiles described by their size a_i

For the experiments, you may use the following instance:

```
In [ ]: A_x, A_y = 33, 32
a = [1, 4, 7, 8, 9, 10, 14, 15, 18]
n = len(a)

A_sq = A_x * A_y
A = A_x * A_y
tiles_sq = sum(a[i] * a[i] for i in range(n))
A_sq, tiles_sq
```

Output

You should find a positions of the bottom-left corners of the tiles so you fill the entire hole in your bathroom.

Exercise

Implement the ILP model of the problem.

```
In [ ]: import gurobipy as g # import Gurobi module

# model -----
model = g.Model()
```

Visualisation

```
In [ ]: import matplotlib.pyplot as plt
from matplotlib.patches import Rectangle
import matplotlib.cm as cm

fig, ax = plt.subplots()

ax.set_xlim((0, A_x))
ax.set_ylim((0, A_y))
ax.set_aspect("equal")

# positions in form of (x, y)
for i in range(n):
    ax.add_patch(Rectangle(positions[i], a[i], a[i], color=cm.rainbow((i + 0.5)/n)))

plt.show()
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```