

作业 2

2025 年第一学期 COMP9021 课程

1 一般事项

1.1 目标

该作业的目的在于：

- 培养你的问题解决能力；
- 设计并以中等规模的 Python 程序形式实现问题解决方案；
- 根据应用程序的预期行为设计并实现一个接口；
- 将代码组织到类中并实现特殊方法；
- 可能的话，练习使用 `re` 和 `numpy` 模块；
- 可选择练习搜索技术的设计与实现，递归是一种很好的方法。

1.2 提交

您的程序将存储在一个名为 `tangram.py` 的文件中。在开发并测试好程序后，请使用 Ed 进行上传（除非您是在 Ed 中直接编写程序）。作业可以多次提交，以最后一次提交的版本为准。您的作业截止日期为 4 月 28 日上午 11 点 59 分。

1.3 评估

该作业满分为 13 分，另有 5 分的加分。它将针对多个输入进行测试。对于每次测试，自动评分脚本会让您的程序运行 30 秒。

作业可在截止日期后最多 5 天内提交。每延迟一整天，可获得的最高分将减少 5%，最多扣减 5 天。因此，如果学生 A 和 B 提交的作业分别值 12 分和 11 分，且都延迟了两天（即超过 24 小时但不超过 48 小时），那么可获得的最高分将为 11.7 分，所以 A 得 $\min(11.7, 12) = 11.7$ 分，B 得 $\min(11.7, 11) = 11$ 分。

您的程序输出应与所指示的**完全一致**。重要提示：

- **附加分仅适用于在截止日期前提交的程序，不适用于截止日期后 5 天内提交的程序。**

该课程的最终成绩上限为 100 分；加分也不能让你在课程中的得分超过 100 分。

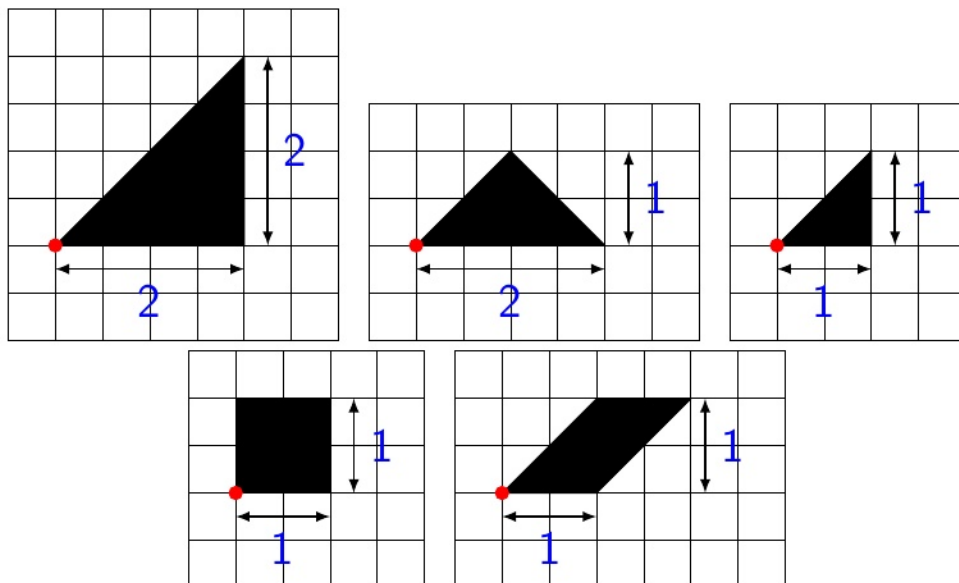
1.4 关于抄袭政策的提醒

您被允许，甚至被鼓励与其他同学讨论解决作业的方法。但讨论的内容必须是算法，而非代码。您必须独立完成代码实现。提交的作业会常规接受检查，以发现学生抄袭并修改他人作业，或在单一实现上密切合作的情况。一旦发现，将予以严厉处罚。

2 七巧板游戏与 TangramTikz 软件包

有关七巧板拼图的概述，请参阅七巧板。所考虑的所有形状均由一个独特的连通组件构成，且内部无孔洞。

我们用两个大三角形、一个中三角形、两个小三角形、一个正方形和一个平行四边形进行游戏，这些图形的尺寸（单位：厘米）如下所示，首先按照图示位置摆放，红色圆点起到“锚点”的作用。



每个部件都可以绕着穿过锚点的垂直线翻转，绕着穿过锚点的水平线翻转，以及绕着其锚点（就像绕着一个支点）旋转 45 度的整数倍，然后再移动，使锚点位于给定的 x 坐标和 y 坐标处。

我们使用 .tex 文件，其中任何一个都可以作为参数传递给 pdflatex 来生成一个 .pdf 文件。

这里有一个关于袋鼠七巧板拼图的 .tex 文件，以及相关的 pdf 文件。

这里有一个关于猫的七巧板拼图的解决方案的 .tex 文件，以及相关的 pdf 文件。

这里有一个关于已解出的“鹅”形七巧板的 .tex 文件，以及相关的 .pdf 文件。

更确切地说，我们使用的 .tex 文件具有以下结构：

```
\documentclass{standalone}
\usepackage{TangramTikz}
\begin{document}
\begin{EnvTangramTikz}
...
\end{EnvTangramTikz}
\end{document}
```

在 `\begin{EnvTangramTikz}` 和 `\end{EnvTangramTikz}` 之间有 7 行内容，每行对应一块拼图，内容分别为：

```
\ PieceTangram [TangSol ](<.....>){...}
```

或者作为

```
七巧板拼图 [拼图解法] <...>(<.....>){...}
```

... 在最后的花括号之间的内容是其中之一

- 对于两个大三角形中的任意一个，记为 `TangGrandTri`。

中号三角形为 `TangMoyTri`。

对于任意两个小三角形中的任意一个，记为 `TangPetTri`。

- `唐卡` 用于广场，以及
- `TangPara` 用于平行四边形。

括号内的两个值分别表示棋子的锚点在可能经过翻转和旋转之后需要移动到的点的 x 坐标和 y 坐标。这些坐标的形式为 `a`、`a+sqrt(2)`、`a-sqrt(2)`、`a+b*sqrt(2)`、`a-b*sqrt(2)`、`sqrt(2)`、`+sqrt(2)`、`-sqrt(2)` 或 `a*sqrt(2)`，其中 `a` 和 `b` 为整数或浮点数，且 `b` 为 0 或严格正数。浮点数的表示形式中包含一个点。当 `a` 为正数时，可以以 `+` 开头，而 `b` 不能。浮点数可以以任意数量的 0 开头，对于绝对值小于 1 的浮点数，小数点前的 0 可以省略（本质上，任何 Python 接受为浮点数字面值的数字序列和一个点都是有效的）。

... 小于号和大于号之间的内容表示一个、两个或三个不同的选项，连续的选项之间用逗号隔开。可能的选项有：

- `xscale=-1`，用于翻转棋子，使左边变为右边，右边变为左边，
- `yscale=-1`，将棋子翻转，使顶部变为底部，底部变为顶部，以及
- `rotate= 旋转 _ 度`，其中 `_` 是 45 的整数倍（正值表示逆时针旋转，负值表示顺时针旋转）。

这些选项可以以任意顺序出现。由于同时进行两次翻转等同于将部件旋转 180 度，所以总是可以使用至多两个选项。`TangramTikz` 包的实现方式是，当一个部件既被旋转又被翻转时，无论给出的选项顺序如何，都会先执行旋转操作，然后再执行翻转操作。

在整个 `.tex` 文件中，标记之间可以有空行和空白字符（空格和制表符）。此外，`%` 符号在一行中最左边的出现标志着从该符号（包括该符号）到物理行末尾（包括 `\n` 字符）为注释。

3 需要执行的任务

您的程序将能够存储在工作目录中的 .tex 文件创建拼图对象，您可以假定这些文件的内容均符合第 2 节中所列出的所有条件。

3.1 报告对每件物品所进行的改造（3 分）

一个“七巧板拼图”对象具有一个“变换”属性，其值是一个字典，包含 7 个键，每个键对应一块拼图。这些键分别是“大三角形 1”、“大三角形 2”、“中三角形”、“小三角形 1”、“小三角形 2”、“正方形”和“平行四边形”。大三角形和小三角形的顺序由它们在 .tex 文件中的顺序决定。更一般地说，字典的键是按照 .tex 文件中相应拼图的顺序创建的。每个键对应的值本身也是一个字典，其键为“x 翻转”、“y 翻转”和“旋转”。对于“x 翻转”和“y 翻转”，至少有一个值为 False，至多有一个值为 True，因为同时进行这两种翻转等同于将拼图旋转 180 度。“旋转”的值是一个介于 0（含）和 360（不含）之间的整数。

以下是一种可能的互动。

3.2 报告顶点的坐标（4 分）

打印出一个“七巧板拼图”对象，会为每个拼图块输出其顶点的坐标。

- 各部件按其最左上角顶点的高度从高到低排列。当两个部件最左上角顶点的高度相同时，位于左侧的部件排在前面。实际上，是从上到下、从左到右扫描平面，当遇到部件的最左上角顶点时，就将该部件列出。
- 坐标采用尽可能简单的形式，如以下示例所示。二进制的“+”和“-”符号两侧均有空格，不存在加 0 的情况，不存在乘以 0 的情况，不存在乘以 1 或 -1 的情况，分数已化简，分母为 1 的分数用整数表示，分数

√

为了避免混淆，在其后跟有 $\sqrt{2}$ 时，这些内容会被括号括起来。
以免人们错误地认为 2 是乘以分母。

对于给定的图形，顶点的枚举从最左上角的顶点开始，按顺时针方向进行。

3.3 创建一个文件来表示已拼好拼图的各个部分（3 分）

“七巧板拼图”类有一个 `draw_pieces()` 方法，该方法接受一个 .tex 文件名作为参数，使用 `tikz` 包而非 `TangramTiks` 包来表示已拼好的七巧板，从而轻松利用前一部分所做的工作。各块的顺序以及给定块的顶点顺序与前一部分相同。平面的原点用一个红点表示。背景中的网格扩展方式为：在最顶点上方至少有一个（5 毫米×5 毫米）的方格，且严格少于两个方格；在最右顶点的右侧至少有一个方格，且严格少于两个方格；在最底顶点的下方至少有一个方格，且严格少于两个方格；在最左顶点的左侧至少有一个方格，且严格少于两个方格。例如：如果图形最右顶点的 x 坐标等于 3.01 或 3.5，则网格向右扩展至 x 坐标为 4；如果图形最右顶点的 x 坐标等于 3.51 或 4，则网格向右扩展至 x 坐标为 4.5。

执行

```
$ python3
...
>>> from tangram import *
```

```
>>> TangramPuzzle('kangaroo.tex').draw_pieces('kangaroo_pieces_on_grid.tex')
>>> TangramPuzzle('cat.tex').draw_pieces('cat_pieces_on_grid.tex')
>>> TangramPuzzle('goose.tex').draw_pieces('goose_pieces_on_grid.tex')
```

分别生成[这个](#)、[这个](#)和[这个](#)的.tex 文件，以及与之相关的[这个](#)、[这个](#)和[这个](#)的.pdf 文件。

建议确保在交互过程中生成的.tex 文件中的空格与所显示的完全一致。不过，在评估时会忽略空白字符，但当然，所有其他非空白字符都必须完全相同，逐字逐句，且所有行的顺序也要一致。

3.4 创建一个文件来表示已解谜题的概要（3 分）

“七巧板拼图”类有一个名为`draw_outline()`的方法，该方法接受一个`.tex`文件名作为参数，通过绘制拼图形状的轮廓（而非每个拼块）来表示已拼好的七巧板，从最左上角的顶点开始，沿顺时针方向绘制。直线段尽可能地在顶点之间延伸。同样，原点用一个红点表示。同样，背景网格的延伸方式使得在最顶点上方、最右顶点右侧、最底顶点下方以及最左顶点左侧至少有一个（5 毫米×5 毫米）的方格，但严格少于两个方格。

执行

```
$ python3
...
>>> from tangram import *
>>> TangramPuzzle('kangaroo.tex').draw_outline('kangaroo_outline_on_grid.tex')
>>> TangramPuzzle('cat.tex').draw_outline('cat_outline_on_grid.tex')
>>> TangramPuzzle('goose.tex').draw_outline('goose_outline_on_grid.tex')
```

分别生成[这个](#)、[这个](#)和[这个](#)的.tex 文件，以及与之相关的[这个](#)、[这个](#)和[这个](#)的.pdf 文件。

再次强调，建议确保在交互过程中生成的.tex 文件中的空格与示例完全一致。不过，在评估时会忽略空格，但当然，所有其他非空格字符都必须完全相同，逐字逐行对应。

3.5 解谜题（5 分加分题）

一个名为`solve_tangram_puzzle()`的函数以文件名作为参数，并返回一个`TangramPuzzle`对象，使得以下交互能够成功：

```
$ python3
...
>>> from tangram import *
>>> solve_tangram_puzzle('kangaroo_outline_on_grid.tex').draw_pieces('solved_kangaroo.tex')
>>> solve_tangram_puzzle('cat_outline_on_grid.tex').draw_pieces('solved_cat.tex')
>>> solve_tangram_puzzle('goose_outline_on_grid.tex').draw_pieces('solved_goose.tex')
>>> ^D
$ diff kangaroo_pieces_on_grid.tex solved_kangaroo.tex
$ diff cat_pieces_on_grid.tex solved_cat.tex
$ diff goose_pieces_on_grid.tex solved_goose.tex
$
```