

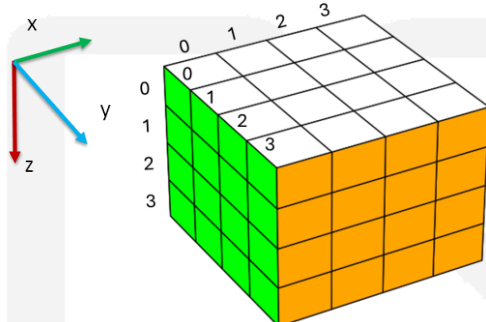
ข้อกำหนด

การเรียกใช้ฟังก์ชันเพื่อการทดสอบ ต้องอยู่ภายใต้เงื่อนไข `if __name__ == '__main__':` เพื่อให้สามารถ import ไปเรียกใช้งานจาก Script อื่น ๆ ได้อย่างเป็นมาตรฐาน

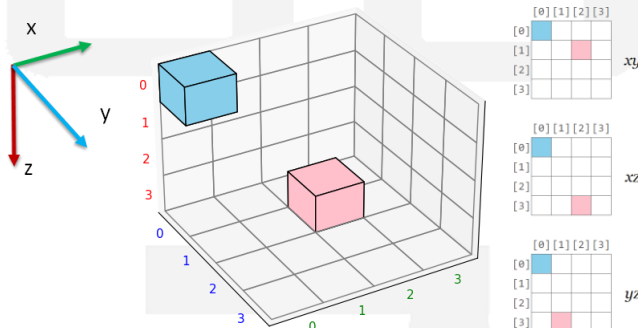
File Header (กรณีไม่เขียน Header จะเสียคะแนน 5%)

```
#!/usr/bin/env python3
# ชื่อ (ไม่ต้องใส่นามสกุล)
# รหัสสนศ
# Sec00x
```

- 1) **100 คะแนน** (Q4P1_6XXXXXXX.py) คุณได้รับลูกบาศก์โปร่งใสขนาด $n \times n \times n$ ($0 < n \leq 300$) แต่ละเซลล์ในลูกบาศก์นี้มีการกำหนดพิกัดในระบบแกน x, y และ z ด้วยค่า (x_i, y_i, z_i) ตามลำดับ เมื่อ $0 \leq x_i, y_i, z_i < n$ รูปด้านล่างแสดงลูกบาศก์ที่ $n = 4$ และแนวแกน x, y และ z ที่อ้างอิง

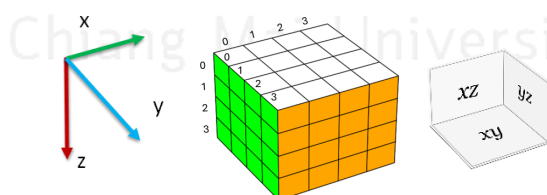


ลูกบาศก์ที่ได้รับจะมีบางเซลล์ที่ทึบแสงระบุด้วย List ของ Tuple ของพิกัดเซลล์ในรูป (x_i, y_i, z_i) เมื่อ $0 \leq x_i, y_i, z_i < n$ เช่นในรูปด้านล่าง เซลล์ที่ทึบแสงคือ $(0,0,0)$ ที่แสดงด้วยสีฟ้า และ $(2,1,3)$ ที่แสดงด้วยสีชมพู ซึ่งจะแทนด้วย List $[(0, 0, 0), (2, 1, 3)]$



หน้าที่ของคุณคือ ให้เขียนฟังก์ชัน

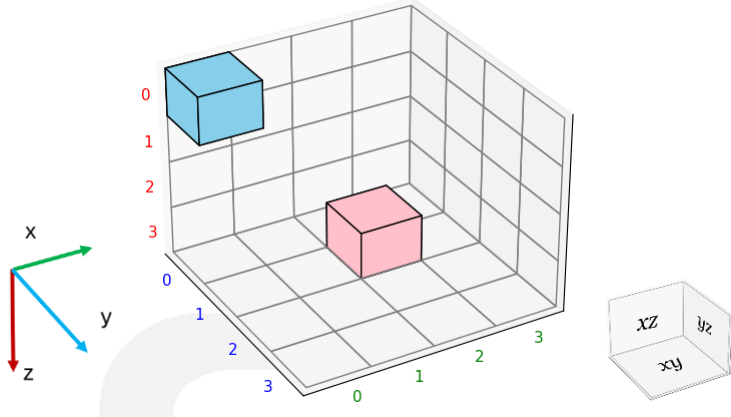
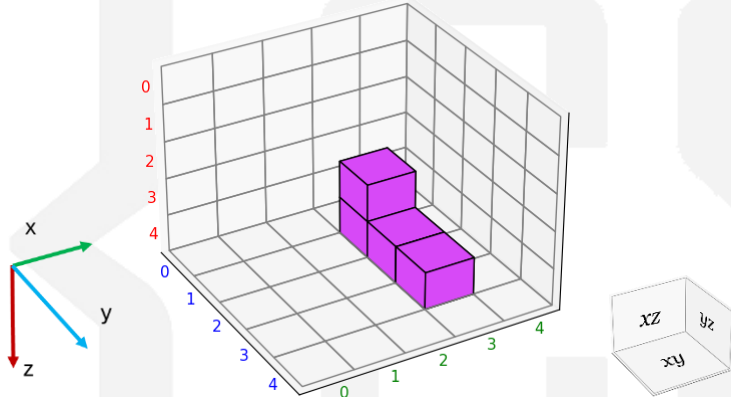
`projections(n: int, opaque_cells: list[tuple[int]]) -> dict[str, list[list[int]]]` เพื่อคืนค่า Dictionary แทนผลลัพธ์การคำนวณหาภาพฉายสองมิติ (Projection) ของลูกบาศก์โปร่งแสงขนาด $n \times n \times n$ ตามที่อธิบายด้านบน ลงบนระนาบทั้งสามเมื่อ `opaque_cells` เป็น List ของ Tuple แทนพิกัดของเซลล์ที่ทึบแสงที่สามารถเป็น List ว่างได้ ทั้งนี้จะไม่มี Coordinate ที่ซ้ำใน `opaque_cells`



ระนาบทั้ง 3 ได้แก่ xy เมื่อมองจากด้านสีขาวในรูป, xz เมื่อมองจากด้านสีส้มในรูป และ yz เมื่อมองจากด้านสีเขียวในรูป โดย Dictionary ที่คืนค่าจะมี key 3 ค่าเท่านั้น ได้แก่ String ' xy ', ' xz ' และ ' yz ' โดยที่แต่ละ key จะมี value เป็น List 2 มิติขนาด $n \times n$ ซึ่งเป็นภาพฉายบนระนาบที่ระบุ กำหนดให้แต่ละเซลล์ใน List 2 มิติมีค่าเป็น 1 หากทึบแสง และ 0 หากโปร่งแสง โดยกำหนดให้ column ที่ 0 และ row ที่ 0 แทนพิกัด $(0,0)$ ในระนาบนั้นๆ

Function Call

Returned Value

 <p>projections(4, [(0, 0, 0), (2, 1, 3)])</p>	<pre>{ 'xy': [[1, 0, 0, 0], [0, 0, 1, 0], [0, 0, 0, 0], [0, 0, 0, 0]], 'xz': [[1, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 1, 0]], 'yz': [[1, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 1, 0, 0]] }</pre>
 <p>projections(5, [(3, 1, 3), (3, 1, 4), (3, 2, 4), (3, 3, 4)])</p>	<pre>{ 'xy': [[0, 0, 0, 0, 0], [0, 0, 0, 1, 0], [0, 0, 0, 1, 0], [0, 0, 0, 1, 0], [0, 0, 0, 1, 0]], 'xz': [[0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 1, 0], [0, 0, 0, 1, 0]], 'yz': [[0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 1, 0, 0, 0], [0, 1, 1, 1, 0]] }</pre>

คำอธิบาย Test Case

มีทั้งหมด 10 Test Case คิดคะแนนเป็น 10 คะแนนต่อ Test Case

- Test Case ที่ 1: เหมือนกรณีตัวอย่าง
 - ต้องมีการคำนวณผลลัพธ์ตามข้อกำหนดของโจทย์ ไม่อนุญาต ให้ Hard Code หรือ ลอก Output จากตัวอย่างมาคั่นค่าหรือแสดงค่า (ถือเป็นการทุจริต)
- Test Case ที่ 2 – 5: n จะมีค่าเป็น 4 เท่านั้น
 - Test Case ที่ 2: มีเซลล์ที่ทึบแสงเพียงเซลล์เดียว
 - Test Case ที่ 3: ตรวจสอบความถูกต้องแค่ระนาบ 'xy'
 - Test Case ที่ 4: ตรวจสอบความถูกต้องแค่ระนาบ 'xz'
 - Test Case ที่ 5: ตรวจสอบความถูกต้องแค่ระนาบ 'yz'
- Test Case ที่ 6 – 10: ทดสอบตามทุกข้อกำหนดของโจทย์

Python Tutor Visualizer: <http://10.4.28.251/tutor/visualize.html>

Grader: <http://10.4.28.251>