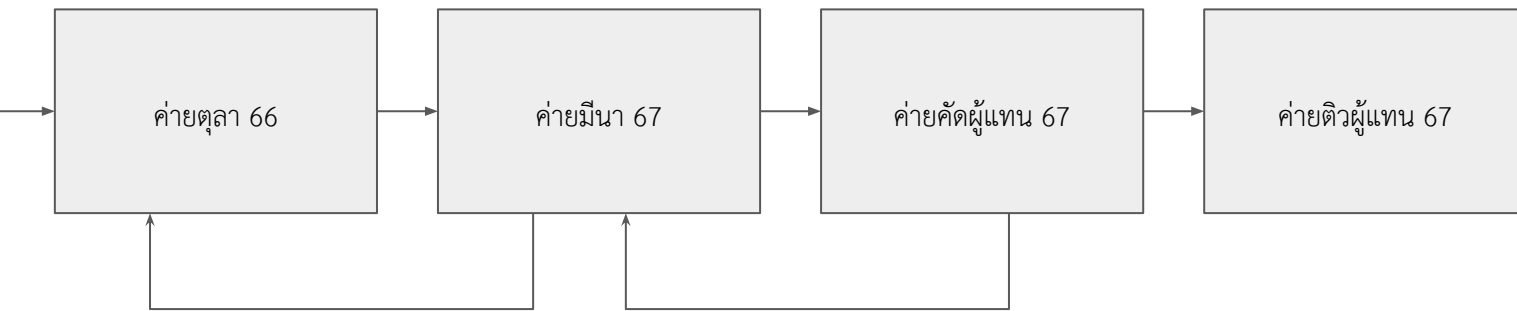


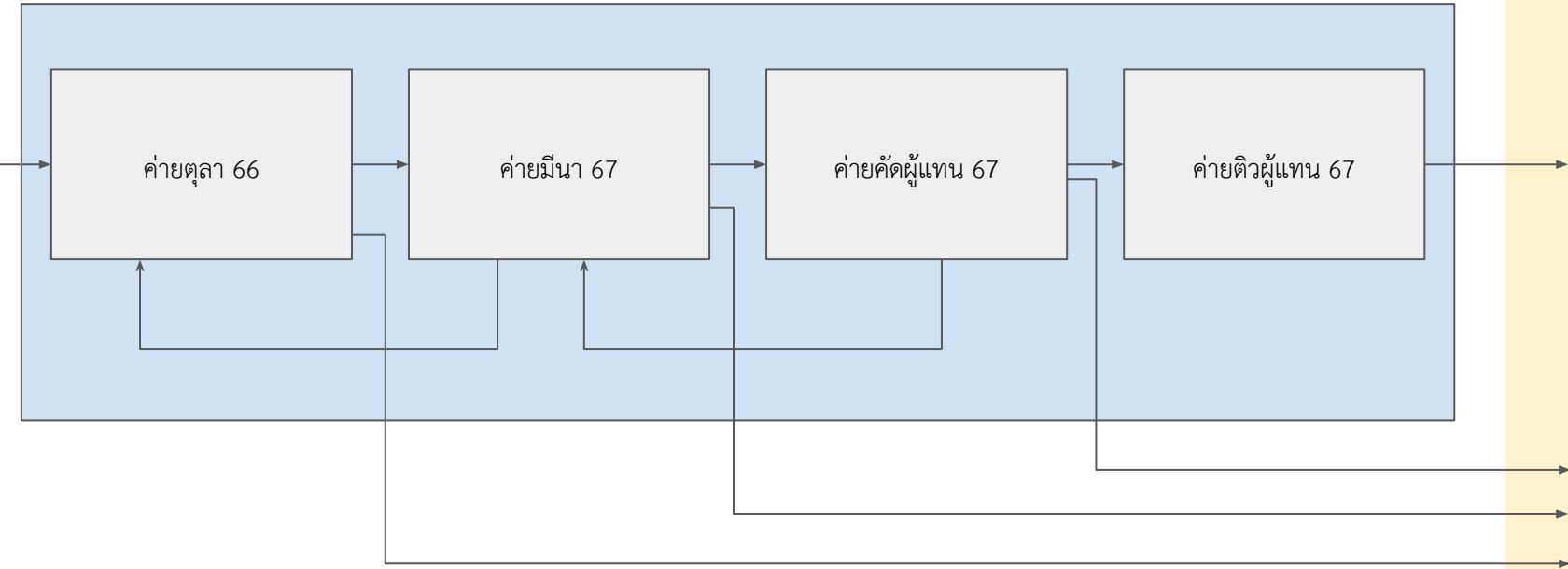
Problem Design

ตั้งโจทย์

ทำไมผมถึงอยากให้คุณได้ลองแต่งโจทย์?



ค่าย สวท.





THAILAND TASK TEAM



(และอื่น ๆ อีกมากมาย)

เป้าหมาย

~~แต่งโจทย์ คนละ 1 ข้อ (พร้อมเฉลย)~~
รู้จักวิธีและขั้นตอนพื้นฐานของการแต่งโจทย์

กำหนดการวันนี้

20.30 - 21.00 พูดคุยเรื่องการแต่งโจทย์

21.00 - 21.15 การเจนเทสเคสด้วยเครื่องมือพื้นฐาน (Python/C++)

21.15 - 22.00 การเจนเทสเคสด้วย Polygon และการสาธิต

22.00 - 23.00 (optional) ทดลองแต่งโจทย์ และเจนเทสเคสตัวเอง

โจทยที่ดี ควรเป็นอย่างไร?

อะไรคือปัจจัยที่จะบอกได้ว่า โจทย์ข้อไหนดี หรือไม่ดี?

ปัญหาการแข่งขันคอมพิวเตอร์

ลองดูแนวทางการเสนอโจทย์ของ IOI

Competition Task

IOI tasks are typically focused on the design of efficient, correct algorithms.

Input and output are to be kept as simple as possible.

The IOI needs both simple and hard tasks that are creative,

so if some of your designed tasks seem easy but creative please submit them too.

However, the nature of previous tasks should not constrain the design of new tasks;

the submission of novel task types not yet seen in IOIs are encouraged.

We are particularly interested in tasks whose basic rules (if not optimal strategy)

are accessible to a wide audience, and tasks that illustrate algorithms and

computational problems that arise in a variety of human endeavours.

Open-ended tasks, ones that do not necessarily have a known efficient or optimal solution,

are welcome.

We are also particularly interested in tasks that go beyond the typical format

in which a program collects input, performs some computation, and returns output.

Examples include “reactive” and “output only” tasks which have been used occasionally

in previous IOIs. Tasks with some measure of solution effectiveness

other than CPU time consumption are encouraged.

A syllabus listing the material generally considered acceptable for IOI tasks

can be found at <https://ioinformatics.org/page/syllabus/12>.

The syllabus is not meant to be restrictive, but to serve as a guideline for task preparation.

To ensure a fair and interesting competition for all, tasks should satisfy the following conditions:

- the tasks should not have been seen by any potential IOI 2023 contestants;
- the tasks should not have been used in any recent similar competition;
- the tasks should be solvable by IOI competitors during an IOI contest round;
- the task descriptions should be unambiguous and easy to understand;
- the tasks should be original and/or innovative.

ที่มา: <https://ioi2023.hu/call-for-tasks/>

IOI Syllabus

<https://ioinformatics.org/files/ioi-syllabus-2023.pdf>
<https://algo.sk/ioi-syllabus/>

<https://sandbox.graders.me/>

(น่าจะใช้งานได้แล้ว)

Test Cases Generation

Basics and Concepts

A+B

```
5
1 2
9 10
4 1
0 5
ABX 8DD
```

```
3
19
5
5
// bad input
```

สิ่งที่เราไม่อยากให้เกิดขึ้น

1. โจทย์ผิด (ร้ายแรงที่สุด)
 - 1.1. โจทย์ตั้งเงื่อนไขอะไรบางอย่าง แต่มันเป็นไปได้ที่จะสร้างอัลกอริทึมมาแก้ภายในขอบเขตที่กำหนด
 - 1.2. โจทย์กล่าวสิ่งที่ขัดแย้ง หรือรับประกันเหตุการณ์ที่เป็นไปไม่ได้ ภายในตัวโจทย์
2. เทสเคสผิด (ร้ายแรงรองลงมา)
 - 2.1. ผิดไปจากข้อกำหนดที่กำหนด เช่นบอกไว้ว่า $N \leq 100,000$ แต่ของจริงมีค่ามากกว่านั้น
3. เทสเคสห่วย (ร้ายแรงมากเช่นกัน)
 - 3.1. โจทย์รับประกัน $N \leq 100,000$ แต่เทสเคสมีแค่กรณี $N \leq 10$ เท่านั้น
4. โจทย์ห่วย (แล้วแต่ความเห็นของแต่ละคน; ถัดไปคือความเห็นส่วนตัวของผม)
 - 4.1. โจทย์ถามสิ่งที่ไม่เกี่ยวข้องกับ competitive programming
 - 4.2. โจทย์จำเป็นต้องให้ผู้เข้าแข่งขันใช้ความรู้ระดับสูง ทำยาก ทำแปลก ทำพิสดาร
 - 4.3. โจทย์เขียนยืดยาว เขียนให้อ่านแล้วงง เขียนไม่ชัดเจน

Simplest Solution

Manual Test Cases (Paper and Pen)

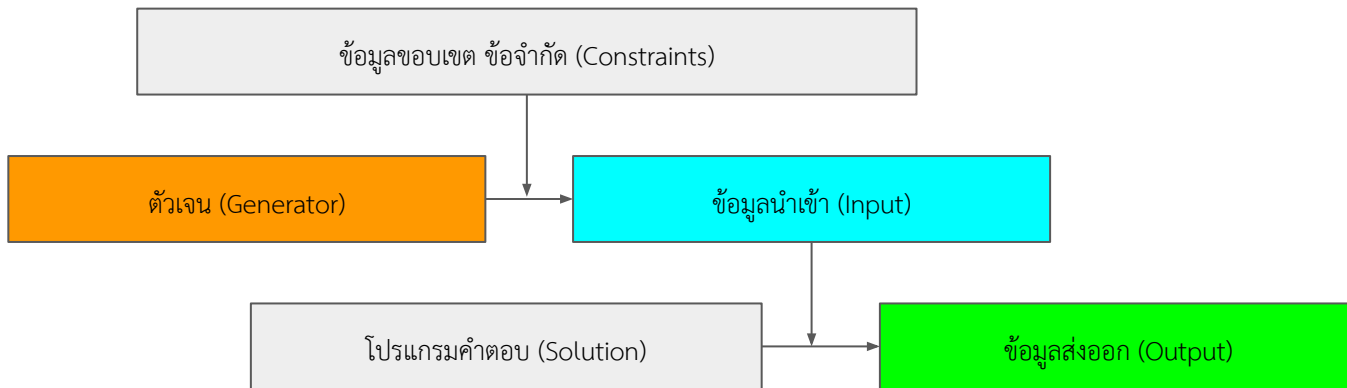
แล้วถ้ามันยุ่งยากกว่านั้นล่ะ

ตัวอย่างการใช้ Python 3 สร้างข้อมูลทดสอบ

```
import random
a, b = map(int, input().split())

if a == 18:
    x = random.randint(4.5*10**a//1, 5*10**a)
else:
    x = random.randint(1, 10**a)
if b == 18:
    y = random.randint(4.5*10**b//1, 5*10**b)
else:
    y = random.randint(1, 10**b)
print(x, y)
```

โมเดลการเจนเทสเคสแบบง่าย ๆ



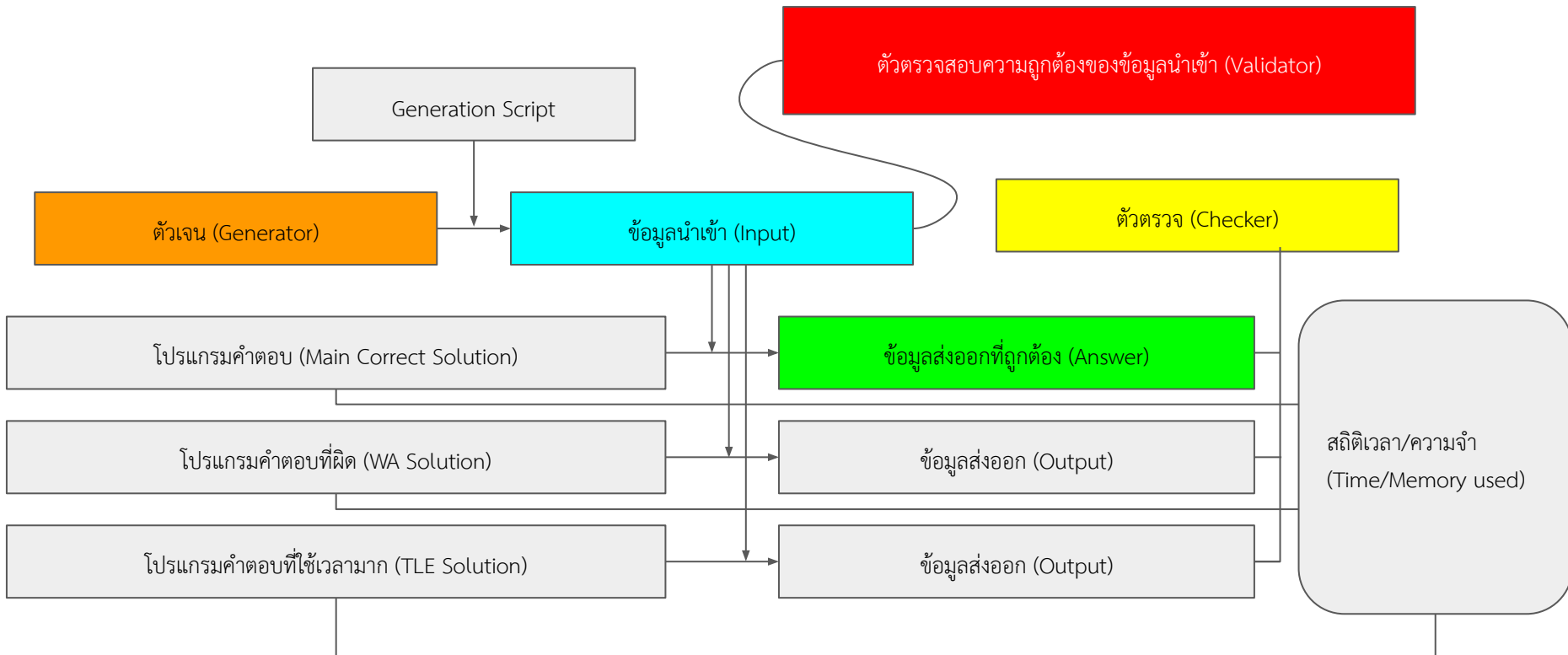
Polygon

<https://polygon.codeforces.com/>

โมเดลการเจนเทสเคสของ Polygon

*WA = Wrong Answer

TLE = Time Limit Exceeded



Graph Generation

ตัวอย่างการเจนกราฟ

aquagenlib.h

```
#include <cstdio>
#include <vector>
#include <unordered_set>
#include <utility>
#include "testlib.h"

using namespace std;

using ll = long long;
using pii = pair<int, int>;

/**
 * Union-find Disjoint Set with Path Compression
 */
class disjoint_set {
private:
    int size;
    ... (209 more lines omitted)
```

ขอขอบคุณสำหรับคุณ Aquablitz11 ที่ได้สร้างและ
แจกจ่าย aquagenlib.h



การสาริตตัวอย่าง