7_鴨鴨收集 (Duck_Collection)

(15 分)

時間限制: 1 second

記憶體限制: 256 MB

題目敘述

YTP 國上有 n 個池塘，每一個池塘內有無限多的鴨鴨，同一個池塘內的鴨鴨種類都是一樣的。

YTP 國的鴨鴨管理者「呱呱」在某一天決定要來玩一個收集鴨鴨的遊戲，這一個遊戲的目標是要收集到所有種類的鴨鴨，而由於有 n 個池塘，因此鴨鴨的種類當然也是 n 種。

這個遊戲每回合都可以發起一次請求，每一次請求可以選擇兩個池塘（可以重複）並在這兩個池塘中各拿一隻鴨鴨出來，而如果你選擇的池塘分別為 (i, j)，那麼這一次請求的花費就會是 $c_{i,j}$，此外，這項遊戲注重先後順序，因此先拿池塘 i 再拿池塘 j 可能會與倒過來拿得到不一樣的花費，也就是說 $c_{i,j} = c_{j,i}$ 不一定成立。

我們的目標除了收集到所有種類的鴨鴨之外，我們也希望花費越少越好，請你設計一個程式計算出最小花費。

輸入格式

第一行輸入一個正整數 n，代表鴨子的種類數以及池塘的個數。

接下來共有 n 行，每一行依序輸入 n 個正整數，第 i 行的第 j 正整數為 $c_{i,j}$，代表你在一回合中先拿了鴨子 i 再拿了鴨子 j 的花費。

輸出格式

請輸出一行，包含一個整數，代表「呱呱」要收集到所有鴨子所需要的最小花費。

資料範圍

- $1 \leq n \leq 18$。
- $1 \leq c_{i,j} \leq 10^9$。

測試範例

輸入範例 1

```
3
10 9 10
4 5 10
7 5 1
```

輸出範例 1

```
5
```

輸入範例 2

```
3
1 10 10
10 1 10
10 10 1
```

輸出範例 2

```
3
```

輸入範例 3

```
3
1 1 1
10 10 10
10 10 10
```

輸出範例 3

```
2
```

範例說明

在範例 1 中，我們各回合的拿法如下：

- 第一回合拿取第 2、1 種鴨子，花費為 4。
- 第二回合拿取兩隻第 3 種的鴨子，花費為 1。
  加起來後得到了總花費為 5。