

Problem H. 貪婪演算法

時間限制: 5 seconds
記憶體限制: 256 megabytes

貪婪演算法 (英語:greedy algorithm)，又稱貪心演算法，是一種在每一步選擇中都採取在當前狀態下最有利的選擇，從而導致結果是最佳的演算法。貪婪法可以解決一些最佳化問題，如：求圖中的最小生成樹、求哈夫曼編碼...對於其他問題，由於貪婪法的高效性以及其所求得的答案比較接近最佳結果，貪婪法也可以用作輔助演算法或者直接解決一些要求結果不特別精確的問題。

剛學完貪婪法的鬼迪在一個大小為 $N \times N$ 的矩陣 a 裡，矩陣的每一格有一個數字 a_{ij} ，且數字為 1 到 N^2 之間的整數，並且每種數字各一個不會重複。鬼迪會選矩陣的任意一個位置出發，並且根據貪婪法，每次都往上下左右最大的數字的格子走且不能超過邊界，如果四周的數字沒有比當前格子更大了則停止移動，鬼迪想知道從哪一格當起點開始移動，可以使得走過的格子數字總和是最大的。

Input

第一行有一個整數 N ($1 \leq N \leq 50$) 接下來的 N 列，每一列有 N 個整數 a_{ij} ($1 \leq a_{ij} \leq N^2$)，代表矩陣第 i 列第 j 行的數字。

保證矩陣內的數字不會重複且 1 到 N^2 每種數字各出現一個。

Output

輸出共兩行

第一行兩個數字為起點的座標，從哪一格當起點可以得到最大總和。定義左上角的位置為 $(1,1)$ ，右下角的位置為 (N,N) 。

第二行為可以得到的最大價值

如果有多組可能的答案，則輸出起始位置的值 a_{ij} 最大的那格。

Examples

standard input	standard output
2 1 2 4 3	1 2 9
3 2 8 5 9 7 4 3 6 1	3 3 23

Note

2	8	5
9	7	4
3	6	1

圖例為範例第二筆測資的最大總和走法