

배달의 부족¹

[문제] 부산대 정컴 출신의 개발자들이 “배달의 민족”보다 더 나은 기술로 무장한 택배 기업 “배달의 부족”을 설립하였다. 이 기업의 핵심 기술은 사용자들의 택배 요구를 매 단위로 받아서 최단 시간에 모두를 서비스할 수 있는 알고리즘에 있다. 각 지역마다 지정된 배달 오토바이는 1 대씩 준비되어 있다. 그리고 음식 F_i 의 배달 주문은 $N \times N$ 그리드(grid) 공간에서 $+i, -i$ 로 표시된다. 여기에서 n 은 주문한 음식점을 의미하고 n 은 그 음식 가 배달되어야 하는 곳, 즉 주문한 사람이 있는 곳을 의미한다.

배달원은 항상 $+a$ 에서 음식을 받아서 $-a$ 로 배달을 해야한다. 그런데 배달 오토바이에는 최대 2개까지 음식을 담을 수 있다. 따라서 아래와 같은 상황에서 V는 $+1$ 으로 가서 F_1 을 받고, 다시 $+2$ 로 가서 F_2 를 받은 다음² -2 로 가서 음식을 내려주고 다시 그 근처 -1 로 가서 음식을 주는 것이 각각을 받아 배달하는 것보다 거리 면에서, 시간 면에서 유리하다. 즉 배달 순서 $[+1, +2, -2, -1]$ 이 한번에 하나씩 배달하는 $[+1, -1, +2, -2]$ 보다 거리 면에서 훨씬 더 나은 배달 순서이다.

10										
9			+2		+1					
8							V			
7		-4			-3					
6										
5		+3						-1		
4										
3						-2				
2			+4							
1										
	1	2	3	4	5	6	7	8	9	10

두 지점 (p, q) 와 (u, v) 를 이동하는 경로의 길이는 맨하튼 거리(Manhattan distance)³로 계산한다. 즉 그 거리는 수직-수평 거리로 계산되어 $|p - u| + |q - v|$ 이다. 위의 예라면 $+1$ 의 위치 좌표가 (5,9)이고 -1 의 위치 좌표가 (8,5)이므로 이 사이의 거리는 $|5-8|+|9-5|=7$ 이다.

¹ 규모가 작은 기업이라 ‘부족’으로 명명함.

² 배달 통에 음식이 가득 찬 상황. 즉 더 이상 추가가 안됨.

³ 맨하튼의 모든 거리는 수직, 수평 성분으로 구성되어 있어서 대각선, 임의의 방향으로 갈 수 없음.

전체 그리드 공간은 1000×1000 으로 고정되어 있다. 배달의 부족 Call Center에는 접수된 주문이 기록되어 있다. 이 주문의 갯수 n 의 범위는 $2 \leq n \leq 10$ 이다. 각 주문 i 는 주문 식당 $+i$ 와 배송지 $-i$ 의 위치를 지정하는 $(+i_x, +i_y)$ 와 $(-i_x, -i_y)$ 의 4개 정수로 주어진다. 그리고 초기 배달 오토바이의 위치는 그리드 지역의 정중앙 (500,500)에 준비되어 있다고 가정한다. 여러분은 이 n 개의 배달을 모두 완성하는 순서를 결정해서 출력해야 한다. 즉 그 식당에서 음식을 pick up한 뒤 집으로 가져다 주는 순서를 $+i -j$ 의 순서로 출력하면 된다.

단 앞서 말한바와 같이 배달 오토바이에는 최대 음식 2개까지 담을 수 있다. 따라서 $n = 4$ 일 때 아래와 같은 방문 순서는 허용되지 않는다. 왜냐하면 +2, +1이 된 상태에서 다시 추가로 +4를 담을 수 없기 때문이다.

[+3, +2, -3, +1, +4, -1, -4, -2]

여러분은 제시된 n 개의 모든 배달을 다 마칠 때 까지 움직인 최소길이의 순서를 찾아서 출력해야 한다. 단 배달은 다 마친 오토바이는 다시 처음 대기 주차장 (500,500)으로 돌아올 필요는 없다. 경우에 따라서 $+i$ 와 $-j$ 의 위치가 같을 수도 있다. 두 사람의 주문이 같은 식당인 경우도 있기 때문이다. 또한 $-i$ 와 $-j$ 도 같을 수 있다. 즉 한 집에서 서로 다른 두 식당의 음식을 한번에 주문할 수도 있기 때문이다.

[입출력] 입력의 첫 줄에는 주문의 갯수 n 이 주어지고 이어지는 n 개의 줄에는 i 의 정보는 나타내는 4개의 정수 a, b, c, d 가 주어진다. 즉 (a, b) 는 $+i$ 의 위치이며 (c, d) 는 $-i$ 의 위치를 나타낸다. 출력으로는 첫 줄에 최적의 방문 순서를 적는다. 그리고 그 다음 줄에 그 총 이동거리를 적는다. 만일 최소 이동 거리가 같은 방문 순서가 하나 이상 있을 경우에는 그 방문 순서 list의 사전식 순서⁴가 가장 빠른 것 하나는 선택해야 한다.

[예제]

입력 stdin	출력 stdout
2 250 250 750 750 750 250 250 750	1 2 -1 -2 2000
2 250 250 750 250 750 750 250 750	1 -1 2 -2 2000

⁴ python list의 “크기” 비교 연산으로 그 순서를 알아낼 수 있다. 예를 들어 $[1,2,3] > [1, 1, 9, 8]$ 은 결과는 True이다.

[제한조건] 프로그램의 이름은 pa09_delivery.{py,c,cpp,java}이다. 제출 횟수는 최대 15번이며 수행시간은 각 test case당 제한 시간은 10초를 초과할 수 없다. 허용가능 코드의 최대 크기는 5,000 bytes 이다. 문제 풀이 1차 마감시간은 2023년 6월1일 24:00 이다. 2차마감은 하루 연장된다. 제출한 프로그램에 대한 풀이(방법과 코드설명)를 작성하여 2023년 6월3일 24:00까지 NESPA “설명게시판”에 제출해야 한다. 제출한 프로그램 풀이과정은 마감이 지나면 공개된다