Clasificatorio Abierto 2025 - Soluciones

Camino

Autor: Pablo Sáez Reyes

Para explicar la solución del problema voy a dar antes varias observaciones:

Observación 1: Podemos descomponer el problema en los ejes Este-Oeste y Norte-Sur, que son independientes. Combinar los resultados para los ejes es sencillo, luego me centraré en el eje Este-Oeste por ejemplo.

Observación 2: El tener que moverse al menos una unidad de distancia cuando desconocemos cuanto nos hemos movido es molesto. Si descomponemos los tramos de longitud desconocida en una parte fija de uno y otra con longitud ≥ 0, entonces es más sencillo de manejar. En adelante supondré que es así.

Observación 3: El órden de los movimientos es irrelevante, supondré que primero muevo las longitudes conocidas y luego las desconocidas.

Usando las observaciones usando los datos puedo obtener la posición en la que me encuentro tras moverme con las cantidades conocidas.

A la hora de ver si puedo regresar con los movimientos de longitud desconocida hay tres casos posibles.

- Me encuentro a la izquierda (oeste) del punto de partida: Podré llegar a él si y solo si tengo al menos un movimiento de longitud desconocida a la derecha (este).
- Me encuentro a la derecha (este) del punto de partida: Podré llegar a él si y solo si tengo al menos un movimiento de longitud desconocida a la izquierda (oeste).
- Me encuentro en el punto de partida: Siempre puedo volver al punto de partida (moviéndome 0 en los tramos restantes).

Si puedo regresar al orígen veo que la distancia mínima que puedo recorrer con la parte no fija es ir directamente desde donde estoy.

Es decir la distancia mínima total es la distancia recorrida fija más la distancia al punto de partida.

Por último, necesito ver cual es la distancia máxima recorrida.

Si la distancia recorrida no es fija (es la misma para todos los caminos posibles que cumplen las condiciones), la distancia adicional recorrida respecto de la mínima viene de ir a la derecha una distancia extra y luego ir a la izquierda esa misma distancia respecto a las longitudes mínimas. Eso implica que hay al menos un movimiento de longitud desconocida a la derecha y otro a la izquierda. Además podemos ver que si existen esos movimientos, los podemos usar para hacer la distancia recorrida arbitrariamente grande.

Por tanto hemos demostrado que la distancia recorrida es fija (y por tanto igual a la mínima) si y solo si no existen dos tramos ilimitados en direcciones opuestas, y en caso contrario la distancia recorrida es ilimitada.

Para programar esto basta recorrer todos los movimientos $O(n)$ y almacenar el desplazamiento fijo y las direcciones en las que me puedo mover ilimitadamente. El resto es comprobar casos en $O(1)$. La complejidad total es $O(n)$.

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;

signed main()
{
    int t;
    cin >> t;
    while (t--) {
        int n;
        cin >> n;
        int dx = 0;
        int dy = 0;
        bool hn = false;
        bool hs = false;
        bool he = false;
        bool ho = false;
        int rec = 0;
        for (int i = 0; i < n; ++i) {
            char d;
            int l;
            cin >> d >> l;
            if (d == 'N') {
                if (l == -1) {
                    l = 1;
                    hn = true;
                }
                dx += l;
            }
            if (d == 'S') {
                if (l == -1) {
                    l = 1;
                    hs = true;
                }
                dx -= l;
            }
            if (d == 'E') {
                if (l == -1) {
                    l = 1;
                    he = true;
                }
                dy += l;
            }
            if (d == 'O') {
                if (l == -1) {
                    l = 1;
                    ho = true;
                }
                dy -= l;
            }
            rec += l;
        }
        rec += abs(dx)+abs(dy);
        bool pos = (dx == 0 || (dx > 0 && hs) || (dx < 0 && hn)) &&
                   (dy == 0 || (dy > 0 && ho) || (dy < 0 && he));
        if (!pos) {
            cout << "-1 -1";
        }
        else {
            cout << rec << ' ';
            bool inf = (hn && hs || he && ho);
            if (inf) cout << -1;
            else cout << rec;
        }
        cout << '\n';
    }
}
```

Pequeño contexto del problema:

Originalmente propuesto para la regional de Castilla y León.

Este problema está inspirado por los problemas en los que te dan una figura en la que todos sus ángulos son rectos y algunos de los lados (pero no todos), y te piden dar el perímetro. Por ejemplo el problema 5 de la Olimpiada Matemática de 2º de la ESO de 2019.

Esos problemas son un poco más difíciles porque no hay una dirección implícita al ver el dibujo. Recorrerlo en uno de los dos sentidos posibles nos lleva al problema aquí planteado.

Originalmente iba a tratar sobre la longitud de una valla (como el problema mencionado antes), pero la posible autointersección de un recorrido llevó a la interpretación actual.

Una generalización interesante para pensar es: qué pasa si en vez de darme una dirección cardinal me dan un vector cualquiera (incluso en una dimensión cualquiera).