

# TALLER PROGCOMP: TRACK EDD

PREFIX SUM 1D Y 2D

**Gabriel Carmona Tabja**

Universidad Técnica Federico Santa María,  
Università di Pisa

April 8, 2024

# Part I

## PREFIX SUM

## PREFIX SUM

### Problema

Dado un arreglo de  $n$  enteros, determinar la suma entre las posiciones  $i$  y  $j$  ( $1 \leq i \leq j \leq n$ ).

## PREFIX SUM

### Problema

Dado un arreglo de  $n$  enteros, determinar la suma entre las posiciones  $i$  y  $j$  ( $1 \leq i \leq j \leq n$ ).

### Solución Inicial

Fácil, recorro todos los números entre  $i$  y  $j$ , acumulando la suma.

¿Cuál es la complejidad?

## PREFIX SUM

### Problema

Dado un arreglo de  $n$  enteros, determinar la suma entre las posiciones  $i$  y  $j$  ( $1 \leq i \leq j \leq n$ ).

### Solución Inicial

Fácil, recorro todos los números entre  $i$  y  $j$ , acumulando la suma.

¿Cuál es la complejidad?  $O(n)$

Nada mal :)

## PREFIX SUM

### Problema

Dado un arreglo  $a$  de  $n$  enteros y  $q$  **queries**, determinar la suma entre las posiciones  $i_q$  y  $j_q$  ( $1 \leq i_q \leq j_q \leq n$ ).

## PREFIX SUM

### Problema

Dado un arreglo  $a$  de  $n$  enteros y  $q$  **queries**, determinar la suma entre las posiciones  $i_q$  y  $j_q$  ( $1 \leq i_q \leq j_q \leq n$ ).

### Solución Inicial

Fácil, por cada query recorro todos los números entre  $i_q$  y  $j_q$ , acumulando la suma.

¿Cuál es la complejidad?

# PREFIX SUM

## Problema

Dado un arreglo  $a$  de  $n$  enteros y  $q$  **queries**, determinar la suma entre las posiciones  $i_q$  y  $j_q$  ( $1 \leq i_q \leq j_q \leq n$ ).

## Solución Inicial

Fácil, por cada query recorro todos los números entre  $i_q$  y  $j_q$ , acumulando la suma.

¿Cuál es la complejidad?  $O(n^2)$

Lo perdimos todo :(.



## PREFIX SUM

### Definición

$$\text{prefix}(i) = \sum_{j=1}^i a_j$$

**OJO:** asumiendo que los índices del arreglo van de 1 hasta  $n$ .

## PREFIX SUM

### Definición

$$\text{prefix}(i) = \sum_{j=1}^i a_j$$

**OJO:** asumiendo que los índices del arreglo van de 1 hasta  $n$ .

### Propiedad

$$\text{suma}(i, j) = \text{prefix}(j) - \text{prefix}(i - 1)$$

### Demostración

$$\text{prefix}(j) = a_1 + a_2 + \dots + a_j$$

$$\text{prefix}(i - 1) = a_1 + a_2 + \dots + a_{i-1}$$

$$\text{prefix}(j) - \text{prefix}(i - 1) = a_1 + a_2 + \dots + a_j - (a_1 + a_2 + \dots + a_{i-1})$$

$$\text{prefix}(j) - \text{prefix}(i - 1) = a_i + \dots + a_j = \text{suma}(i, j)$$

## CÓDIGO

```
1 struct prefix {  
2     vector< int > p;  
3     prefix(vector< int > &nums) {  
4         p.push_back(0);  
5         for(int i = 0; i < nums.size(); i++) {  
6             p.push_back(nums[i] + p[i]);  
7         }  
8     }  
9     int query(int i, int j) {  
10        return p[j] - p[i - 1];  
11    }  
12 }
```

Complejidad de construcción:  $O(n)$

Complejidad por query:  $O(1)$  :)

## LIMITACIONES Y DETALLES

- ▶ Para que funcione la operación debe tener inverso
- ▶ Si hay updates, lo perdemos todo :(ul>- Lo resolveremos con algo mágico más adelante

## Part II

### PREFIX SUM 2D

## PREFIX SUM 2D

Ahora extenderemos la idea de Prefix Sum a dos dimensiones.

10	20	30
5	10	20
2	4	6

Determinar la suma en la sub matrix  $i, j, k, l$ , lo que significaría la suma de lo que esta entre la fila  $i$  hasta la  $j$  y la columna  $k$  hasta la  $l$ .

### Ejemplo

$$\text{suma}(2, 3, 2, 3) = 40$$

**OJO:** asumiendo que los indices están entre 1 y  $n$ .

## PREFIX SUM 2D

La primera fila y la primera columna son fáciles de rellenar.

10	20	30
5	10	20
2	4	6

10	30	60
15	?	?
17	?	?

Ahora, ¿cómo rellenamos el resto?

## PREFIX SUM 2D

La primera fila y la primera columna son fáciles de rellenar.

10	20	30
5	10	20
2	4	6

10	30	60
15	?	?
17	?	?

Ahora, ¿cómo rellenamos el resto?

La intuición sería que a la posición  $i, j$  se le sume lo de  $j - 1$  y  $i - 1$



## PREFIX SUM 2D

La primera fila y la primera columna son fáciles de rellenar.

10	20	30
5	10	20
2	4	6

10	30	60
15	?	?
17	?	?

Ahora, ¿cómo rellenamos el resto?

La intuición sería que a la posición  $i, j$  se le sume lo de  $j - 1$  y  $i - 1$ . Pero, pasa lo siguiente:

- ▶  $prefix(2, 2) = prefix(1, 2) + prefix(2, 1)$
- ▶  $prefix(1, 2) = prefix(1, 1) + a[1][2]$
- ▶  $prefix(2, 1) = prefix(1, 1) + a[2][1]$

Reptimos  $prefix(1, 1)$ .

## PREFIX SUM 2D

La primera fila y la primera columna son fáciles de rellenar.

10	20	30
5	10	20
2	4	6

10	30	60
15	?	?
17	?	?

Ahora, ¿cómo rellenamos el resto?

La intuición sería que a la posición  $i, j$  se le sume lo de  $j - 1$  y  $i - 1$ . Pero, pasa lo siguiente:

- ▶  $prefix(2, 2) = prefix(1, 2) + prefix(2, 1)$
- ▶  $prefix(1, 2) = prefix(1, 1) + a[1][2]$
- ▶  $prefix(2, 1) = prefix(1, 1) + a[2][1]$

Reptimos  $prefix(1, 1)$ .

Por lo que, la formula final sería:

$$prefix(j, l) = prefix(j - 1, l) + prefix(j, l - 1) - prefix(j - 1, l - 1)$$

## CÓDIGO

```
1 struct prefix_2D {
2     vector< vector< int > > p;
3     prefix_2D(vector< vector< int > > &m) {
4         p.assign(m.size(), vector< int >(m[0].size(), 0));
5         for(int j = 1; j < m.size(); j++) {
6             for(int l = 1; l < m[0].size(); l++) {
7                 p[j][l] = p[j - 1][l] + p[j][l - 1]
8                     - p[j - 1][l - 1] + m[j][l];
9             }
10        }
11    }
12};
```

## CALCULAR LA SUMA

Ahora, queremos permitir la operación  $\text{suma}(i, j, k, l)$ .

0	0	0	0
0	10	30	60
0	15	45	95
0	17	51	107

Calculemos  $\text{suma}(1, 3, 2, 3)$ .

## CALCULAR LA SUMA

Ahora, queremos permitir la operación  $\text{suma}(i, j, k, l)$ .

0	0	0	0
0	10	30	60
0	15	45	95
0	17	51	107

Calculemos  $\text{suma}(1, 3, 2, 3)$ .

$$\text{suma}(1, 3, 2, 3) = \text{prefix}(3, 3) - \text{prefix}(1 - 1, 3) - \text{prefix}(3, 2 - 1) + \text{prefix}(1 - 1, 2 - 1)$$

## CALCULAR LA SUMA

Ahora, queremos permitir la operación  $\text{suma}(i, j, k, l)$ .

0	0	0	0
0	10	30	60
0	15	45	95
0	17	51	107

Calculemos  $\text{suma}(1, 3, 2, 3)$ .

$$\text{suma}(1, 3, 2, 3) = \text{prefix}(3, 3) - \text{prefix}(1 - 1, 3) - \text{prefix}(3, 2 - 1) + \text{prefix}(1 - 1, 2 - 1)$$

Formula general:

$$\text{suma}(i, j, k, l) = \text{prefix}(j, l) - \text{prefix}(i - 1, l) - \text{prefix}(j, k - 1) + \text{prefix}(i - 1, k - 1)$$

# REFERENCES I