

# Estructuras de Datos y Algoritmos II

## Tarea 2

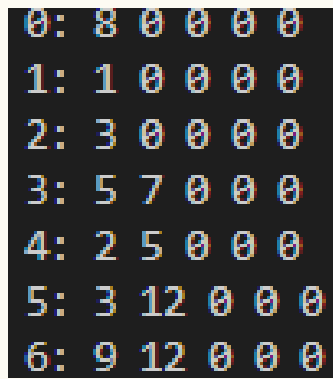
Julián Rodríguez Isiordia

Licenciatura en Computación Matemática

---

### Complejidad

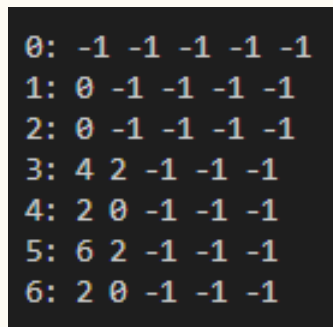
- **DFS:** Realiza un recorrido en profundidad del árbol, visitando cada nodo una vez, por lo que tiene complejidad  $O(N)$ , va registrando en un arreglo aparte la suma total de los nodos con sus ancestros.
- ***sparseTableParent*:** Esta función genera la primera *sparse table* de los padres que nos da las conexiones que tienen los nodos en distancia  $2^i$ , esto tiene complejidad  $O(N \log N)$ , donde  $N$  representa los nodos del árbol, esto ya que itera sobre todos los niveles del árbol, o bien los intervalos de la tabla, y en cada nivel ocupa verificar los nodos para saber si es necesario un cambio.
- ***sparseTable*:** Realiza un procedimiento similar a la función anterior, pero esta en lugar de almacenar el ancestro, almacena las sumas hasta esa distancia  $2^i$ . La complejidad también es  $O(N \log N)$ .
- ***query*:** Realiza una consulta para encontrar la suma de los valores de los nodos en el camino desde el nodo dado hasta su ancestro  $d$ -ésimo. Utiliza la tabla *sparse* para encontrar el ancestro  $d$ -ésimo de manera eficiente, realizando  $O(\log N)$  operaciones, pues busca los intervalos separándolos en potencias de 2.



A sparse table visualization showing the total sum for each node at distances 2^0, 2^1, 2^2, 2^3, 2^4, 2^5, and 2^6. The table has 7 rows and 6 columns. The first column contains node indices from 0 to 6. The subsequent columns contain the sum values. The values are: Row 0: 8, 0, 0, 0, 0, 0; Row 1: 1, 0, 0, 0, 0, 0; Row 2: 3, 0, 0, 0, 0, 0; Row 3: 5, 7, 0, 0, 0, 0; Row 4: 2, 5, 0, 0, 0, 0; Row 5: 3, 12, 0, 0, 0, 0; Row 6: 9, 12, 0, 0, 0, 0.

|    |   |    |   |   |   |
|----|---|----|---|---|---|
| 0: | 8 | 0  | 0 | 0 | 0 |
| 1: | 1 | 0  | 0 | 0 | 0 |
| 2: | 3 | 0  | 0 | 0 | 0 |
| 3: | 5 | 7  | 0 | 0 | 0 |
| 4: | 2 | 5  | 0 | 0 | 0 |
| 5: | 3 | 12 | 0 | 0 | 0 |
| 6: | 9 | 12 | 0 | 0 | 0 |

Figura 1: Sparse table suma total del ejemplo



A sparse table visualization showing the ancestors for each node at distances 2^0, 2^1, 2^2, 2^3, 2^4, 2^5, and 2^6. The table has 7 rows and 6 columns. The first column contains node indices from 0 to 6. The subsequent columns contain the ancestor values. The values are: Row 0: -1, -1, -1, -1, -1; Row 1: 0, -1, -1, -1, -1; Row 2: 0, -1, -1, -1, -1; Row 3: 4, 2, -1, -1, -1; Row 4: 2, 0, -1, -1, -1; Row 5: 6, 2, -1, -1, -1; Row 6: 2, 0, -1, -1, -1.

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| 0: | -1 | -1 | -1 | -1 | -1 |
| 1: | 0  | -1 | -1 | -1 | -1 |
| 2: | 0  | -1 | -1 | -1 | -1 |
| 3: | 4  | 2  | -1 | -1 | -1 |
| 4: | 2  | 0  | -1 | -1 | -1 |
| 5: | 6  | 2  | -1 | -1 | -1 |
| 6: | 2  | 0  | -1 | -1 | -1 |

Figura 2: Sparse table ancestros del ejemplo

Se implementaron dos *sparse table*, una para los ancestros que guarda para cada nodo, cuales son sus ancestros, los  $-1$  representan que no hay nodos a esa distancia y otra tabla que almacena la suma total a distancias  $2^i$ , donde  $i$  representa el índice del nodo correspondiente en la fila. En el ejemplo de Hackerrank se ve de la siguiente forma