

Lowest Common Ancestor

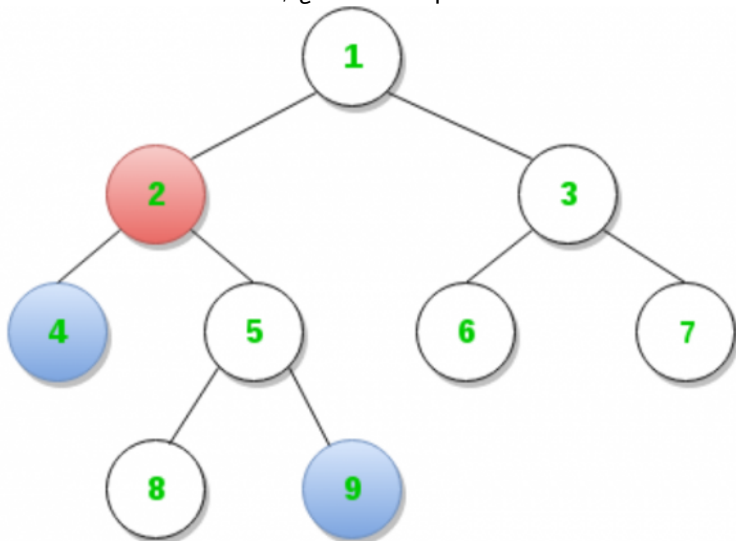
Algoritmo usando RMQ / segment trees

Marco Antonio Gómez Martín

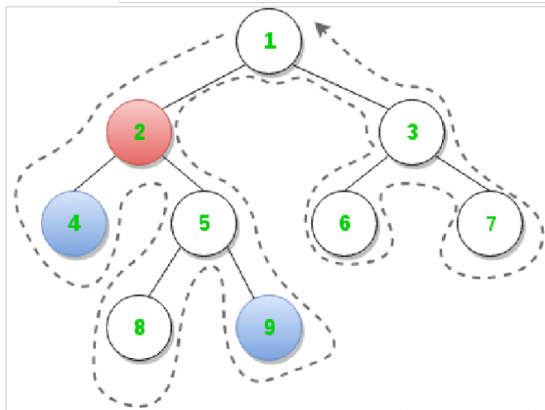
Facultad de Informática - UCM

Lowest Common Ancestor (LCA)

Dados dos nodos de un árbol, ¿cuál es el padre común más cercano?



Lowest Common Ancestor (LCA)



Euler Tour

An euler tour of the tree starting from node 1 will yield:

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 4 | 2 | 5 | 8 | 5 | 9 | 5 | 2 | 1 | 3 | 6 | 3 | 7 | 3 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

The corresponding levels for every node in Euler tour:

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 1 | 2 | 3 | 2 | 3 | 2 | 1 | 0 | 1 | 2 | 1 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

- 1 Hacer un recorrido de Euler del árbol, rellorando:
 - Array con los nodos recorridos (recorrido de Euler), `euler`.
 - Array con las profundidades de cada nodo, `prof`.
 - Array con la primera aparición de cada nodo, `first`.
- 2 Construir el árbol de segmentos sobre `prof`.
- 3 Usando el array `first`, sacar los índices que ocupan los nodos en el tour.
- 4 Obtener el RMQ sobre `prof`. El RMQ debe adaptarse para devolver *la posición* que ocupa el mínimo (no su valor).
- 5 Con la posición del mínimo, ir a `euler` para sacar el id del nodo.

Árboles de Fenwick

```
int idx; // Siguiete entrada en euler y prof
int euler[2 * MAX_V - 1];
int prof[2 * MAX_V - 1]; // Prof. del nodo en euler[] (RMQ sobre él)
int first[MAX_V]; // Primera aparición del nodo i en euler[]

void eulerTour(int u, int parent, int d) { // d = depth

    first[u] = idx; euler[idx] = u; prof[idx] = d; ++idx;

    for (int i = 0; i < adj[u].size(); ++i) {
        int v = adj[u][i];
        if (v == parent) continue;

        eulerTour(v, u, d+1);
        euler[idx] = u; prof[idx] = d; ++idx;
    }
}
```

- ❶ Hacer un recorrido de Euler del árbol, rellenando:
 - Array con los nodos recorridos (recorrido de Euler), `euler`.
 - Array con las profundidades de cada nodo, `prof`.
 - Array con la primera aparición de cada nodo, `first`.
- ❷ Construir el árbol de segmentos sobre `prof`.
- ❸ Usando el array `first`, sacar los índices que ocupan los nodos en el tour.
- ❹ Obtener el RMQ sobre `prof`. El RMQ debe adaptarse para devolver *la posición* que ocupa el mínimo (no su valor).
- ❺ Con la posición del mínimo, ir a `euler` para sacar el id del nodo.

```
int lca(int u, int v) {  
    return euler[st.query(first[u], first[v])];  
}
```