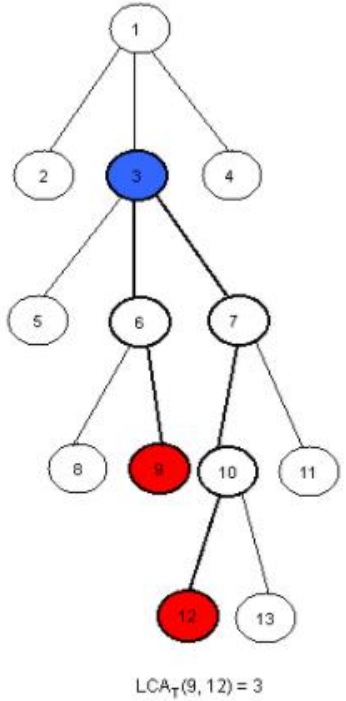
Lowest Common Ancestor

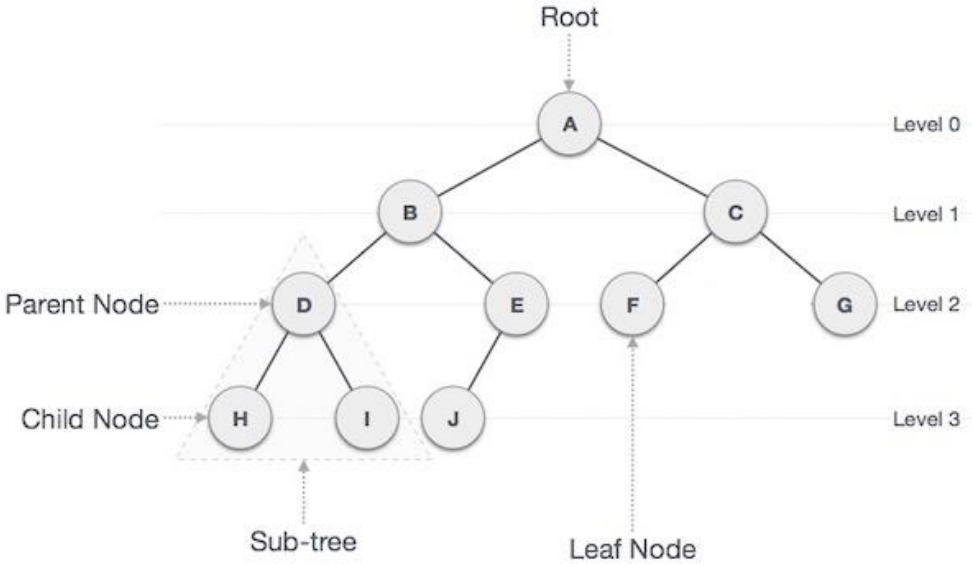
B.S. Rodolfo Mercado Gonzales

Lowest Common Ancestor (LCA)

Dado un árbol $\boldsymbol{G}$ con $\boldsymbol{n}$ nodos, el LCA de dos nodos $\boldsymbol{u}, \boldsymbol{v}$ es el nodo más alejado de la raíz que es un ancestro común para $\boldsymbol{u}$ y $\boldsymbol{v}$.



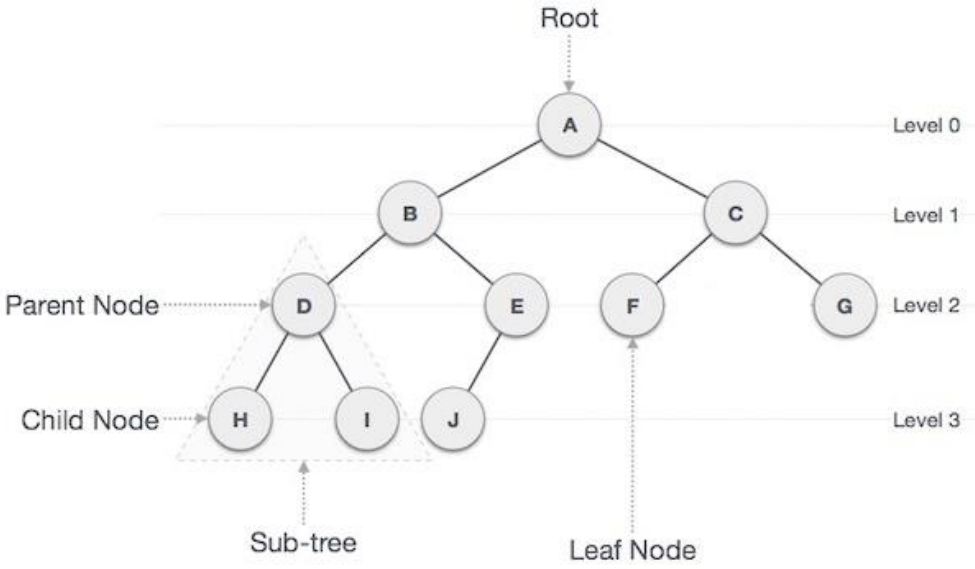
Solución trivial



Caso 1 : u y v se encuentran en el mismo nivel.

Caso 2 : u y v se encuentran en niveles distintos.

Solución trivial



Caso 1 : u y v se encuentran en el mismo nivel.

Subimos a través de los padres de nivel en nivel hasta coincidir

Caso 2 : u y v se encuentran en niveles distintos.

Igualamos el nivel de los nodos y nos queda el caso 1.

-Preprocesamiento : $\boldsymbol{O(n)}$
-Query : $\boldsymbol{O(n)}$

Nivel y padre de nodos

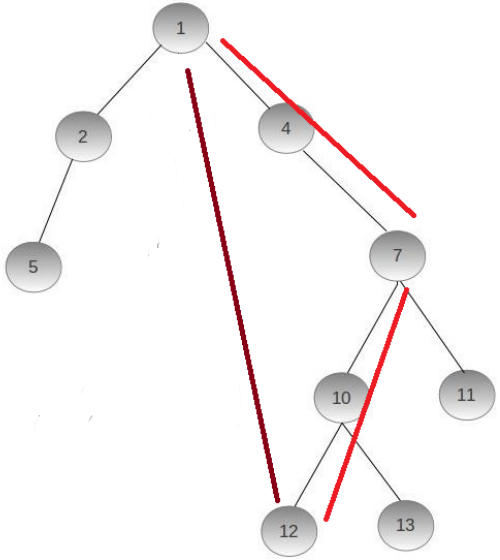
```
int L[ N ], T[ N ]; // T[ root ] = -1, inciar L[] en -1
vector<int> adj[ N ];

void dfs( int u, int nivel ){
    L[ u ] = nivel;
    for( int i = 0; i < adj[ u ].size(); ++i ){
        int v = adj[ u ][ i ];
        if( L[ v ]  == - 1 ){
            T[ v ] = u ;
            dfs( v , nivel + 1 );
        }
    }
}
```

Solución eficiente

Usaremos una matriz para calcular los ancestros a una distancia igual a una potencia de 2.

$$\boldsymbol{anc}[\,\boldsymbol{i}\,][\,\boldsymbol{j}\,] = ancestro\ de\ i\ a\ distancia\ 2^j$$

$$anc[\,i\,][\,j\,] = anc[\,\boldsymbol{anc}[\,\boldsymbol{i}\,][\,\boldsymbol{j-1}\,]\,][\,j-1]$$



Solución eficiente

```
int anc[ N ][ log2( N ) ];

void preproceso(){

    for( int i = 0; i < n; ++i )
        for( int j = 0; ( 1 << j ) < n; ++j )
            anc[ i ][ j ] = -1;

    for( int i = 0; i < n; ++i )
        anc[ i ][ 0 ] = T[ i ];

    for( int j = 1; ( 1 << j ) < n; ++j )
        for( int i = 0; i < n; ++i )
            if( anc[ i ][ j - 1 ] != -1 )
                anc[ i ][ j ] = anc[ anc[ i ][ j - 1 ] ][ j - 1 ];

}
```

$O(n \log n)$

Solución eficiente

```
int lca( int u, int v ){
    // fijaremos que nivel de u es mayor
    if( L[ u ] < L[ v ] ) swap( u, v );

    // log ( L[ u ] )
    int lg = 31 - ( __builtin_clz( L[ u ] ) );

    // ancestro de u al mismo nivel que v
    for( int i = lg; i >= 0; i-- ){
        if( L[ u ] - ( 1 << i ) >= L[ v ] ){
            u = anc[ u ][ i ];
        }
    }
    if( u == v ) return u;

    // subimos ambos al mismo tiempo
    for( int i = lg; i >= 0; i-- )
        if( anc[ u ][ i ] != -1 && anc[ u ][ i ] != anc[ v ][ i ] )
            u = anc[ u ][ i ], v = anc[ v ][ i ];

    return T[ u ];
}
```

$O(\log n)$

Problemas

SPOJ – Lowest Common Ancestor (LCASQ)

SPOJ – Ants Colony

Referencias

- Topcoder, Range Minimum Query and Lowest Common Ancestor

¡ Good luck and have fun !