

# OBSERVACIONES DEL LA PRACTICA

Daniela Espinosa 202022615

Tomás la Rotta 202021354

## Preguntas de análisis

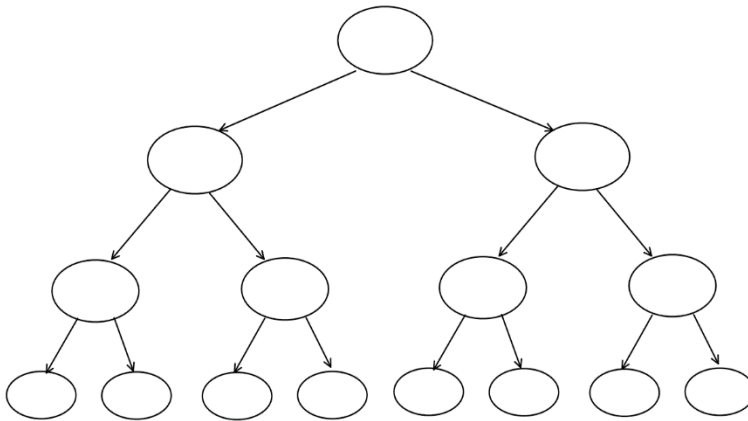
- 1) ¿Qué relación encuentra entre el número de elementos en el árbol y la altura del árbol?

Según la altura del árbol, se puede determinar el número máximo de elementos que este puede tener, con la siguiente ecuación:

$$\text{Número máximo de elementos} = \sum_{n=0}^h 2^n \quad (h: \text{altura del árbol})$$

Ejemplo:

$h = 3$



$$15 = 2^0 + 2^1 + 2^2 + 2^3$$

Además, el número mínimo de elementos que podría tener el árbol sería  $h + 1$ . Por lo que, siguiendo con el mismo ejemplo, el número mínimo de elementos sería 4. Entonces el número de elementos  $N$  cumple con la siguiente regla:

$$h + 1 \leq N \leq \sum_{n=0}^h 2^n$$

Para aprovechar este tipo de estructura, la idea es que:

$$h + 1 \ll N$$

- 2) ¿Si tuviera que responder esa misma consulta y la información estuviera en tablas de hash y no en un BST, cree que el tiempo de respuesta sería mayor o menor? ¿Por qué?

El tiempo de respuesta sería mayor porque en tablas de hash, porque el BST, al estar ordenado, permite que la búsqueda sea mucho más eficiente que como sería con una tabla de hash. Además, aunque la tabla también esté ordenada, el recorrido en el peor caso tiene complejidad de  $O(N)$ , mientras que en un BST la búsqueda se puede hacer en  $\log(N)$  comparaciones.

- 3) ¿Qué operación del TAD se utiliza para retornar una lista con la información encontrada en un rango de fechas?

Se utiliza la operación `values(map, keylo, keylohi)`, que tiene como parámetros el mapa (map), el límite inferior (keylo) y el límite superior (keylohi) de un rango determinado. Esta retorna las llaves que estén dentro de dicho rango, y se usa en la línea 186 como se puede ver en la siguiente imagen:

```
182 def getCrimesByRange(analyzer, initialDate, finalDate):
183     """
184     Retorna el numero de crímenes en un rango de fechas.
185     """
186     lst = om.values(analyzer['dateIndex'], initialDate, finalDate)
187     totcrimes = 0
188     for lstdate in lt.iterator(lst):
189         totcrimes += lt.size(lstdate['lstcrimes'])
190     return totcrimes
```

En este caso el mapa es `analyzer['dateIndex']`, el límite inferior es `initialDate` y el límite superior es `finalDate`.