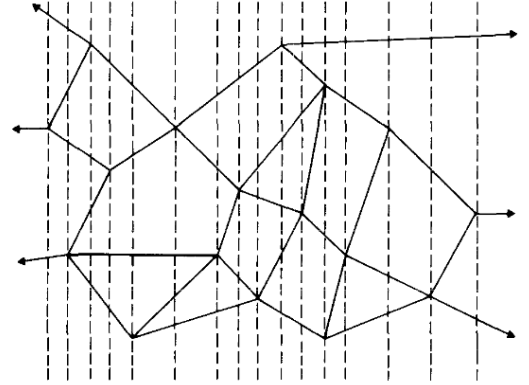
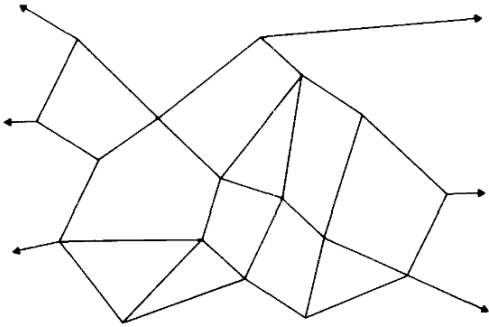


Complejidades (Importantes)

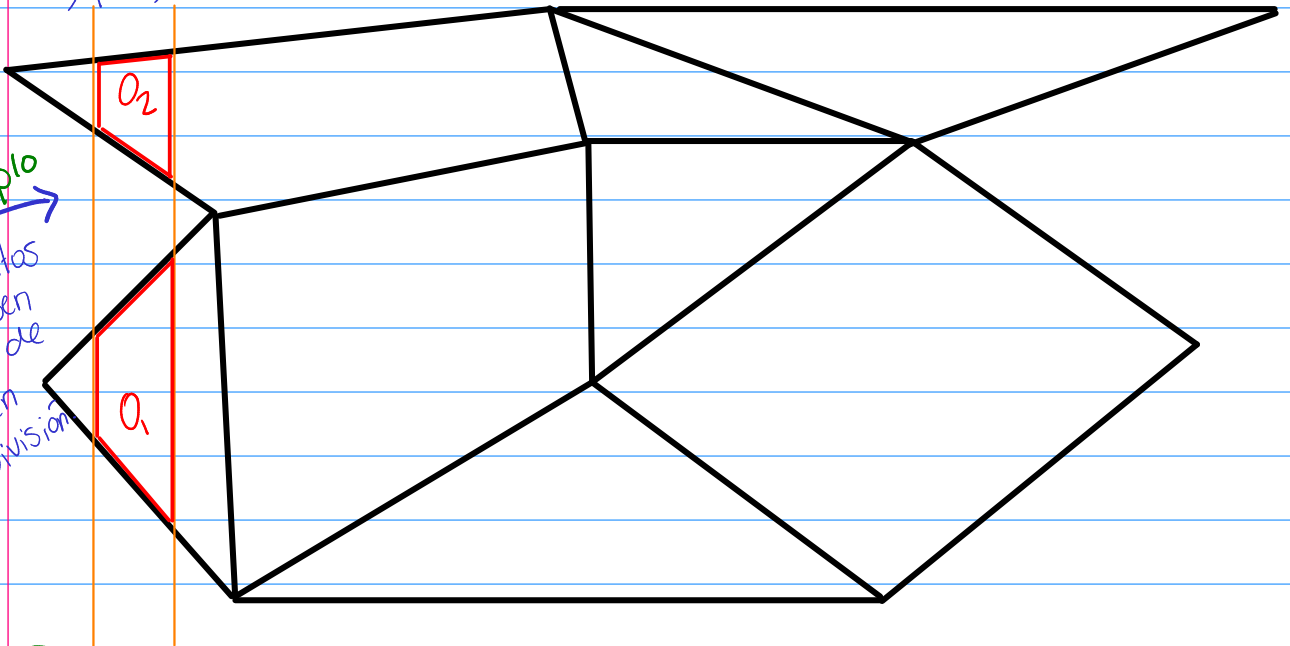
1. Tiempo de preprocesamiento (menos importante).
2. Espacio requerido para almacenar la estructura.
3. Tiempo de consulta.

Construcción Dobkin-Lipton: → Método a usar.



- ① Trazar líneas verticales (Paralelas a Y) desde cada vértice en la subdivisión plana.
- ② El paso anterior divide en losas verticales, esto es

l_1 l_2

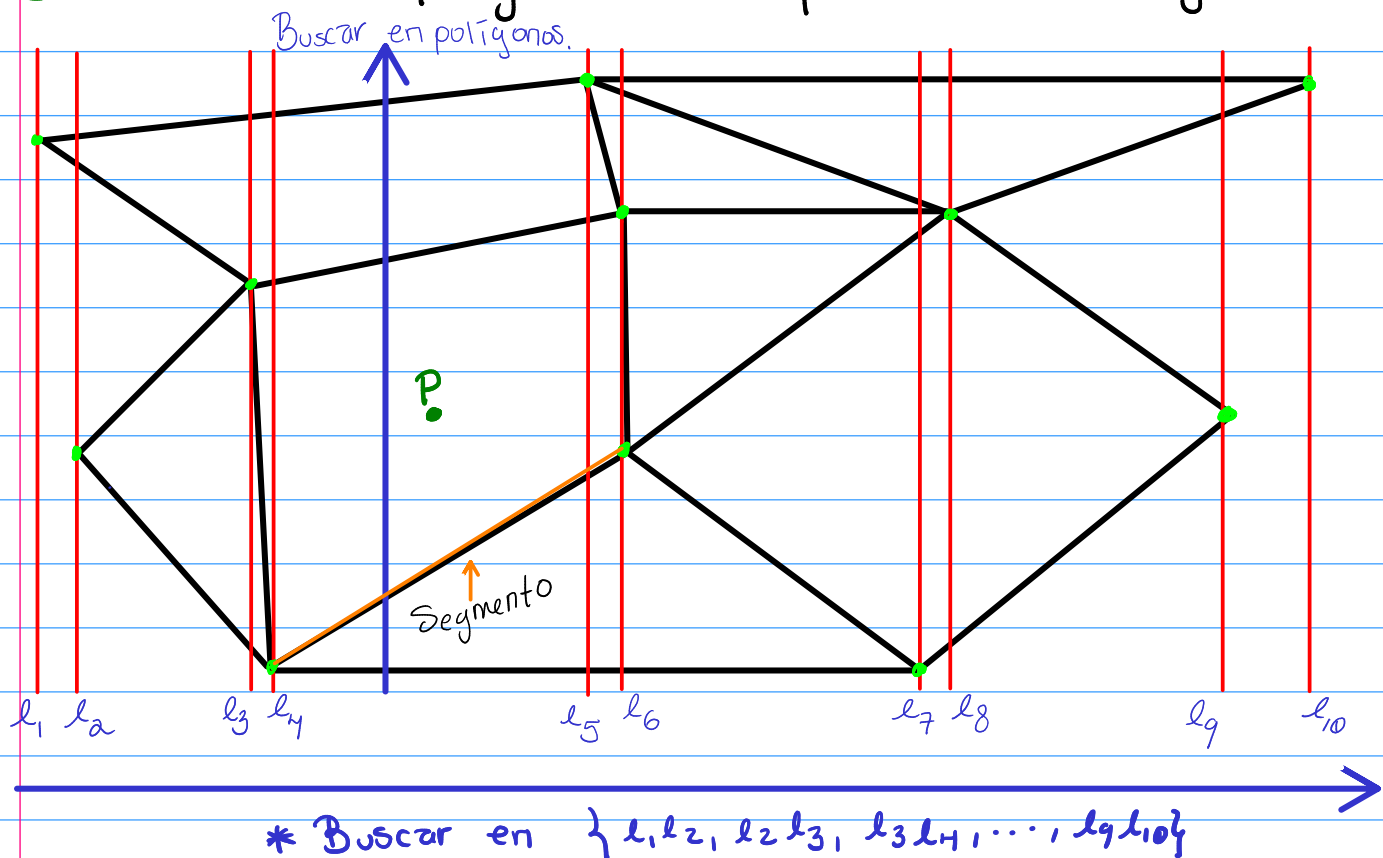


Ficticio
sólo ejemplo →
Los segmentos l_1, l_2 deben depender de los vértices en la subdivisión.

- ③ Los segmentos de línea de la subdivisión que cortan una losa están ordenados (Inferior → Superior).
- ④ Asociamos cada segmento de línea con el polígono encima de este.

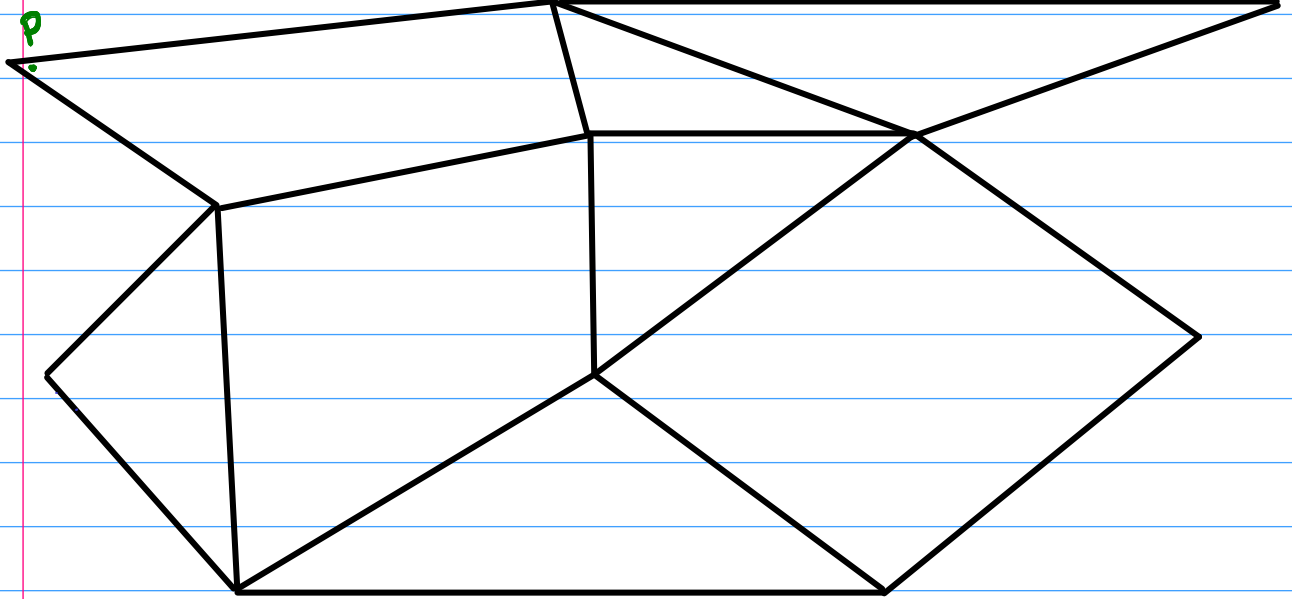
Consultas (Obtener el polígono que contiene el punto).

- ① Realizamos búsqueda binaria en cada losa vertical (sobre x) para encontrar la losa que contiene al punto.
- ② Realizamos búsqueda binaria en la losa para encontrar el segmento más cercano que este debajo del punto.
- ③ Encontramos el polígono al cuál pertenece el segmento.



* ¿Cuánto nos costaría guardar la subdivisión y losas en un árbol de búsqueda binaria? R. / $O(n^2)$, pues en el peor de los casos las losas se intersectan un máximo número de veces.

Encuentra P



Def. Persistencia.

Una estructura de datos es persistente si se pueden realizar modificaciones a esta a manera de que podamos acceder a su versión presente y versiones pasadas.

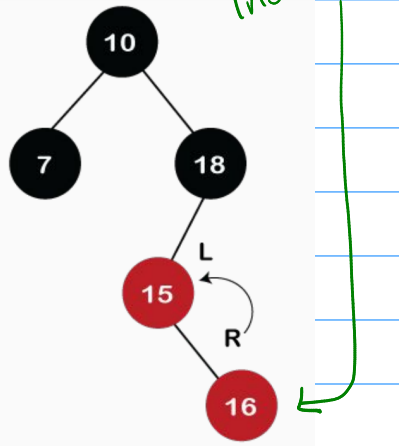
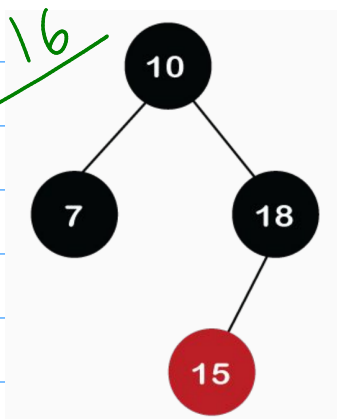
Reducción del problema.

Cocentremos nuestra atención en una sola losa, así el paso de una losa a otra de izq. \rightarrow der. es similar a insertar y eliminar segmentos. Entonces queremos una estructura persistente para realizar la búsqueda de nuestro punto y poder localizar en que polígono se encuentra.

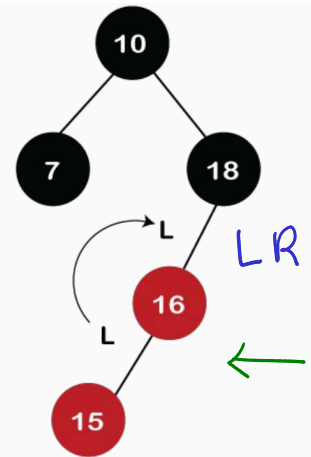
Obs. En otro caso son llamados efímeros.

Rojos — Negros (persistentes)

Insertar 16



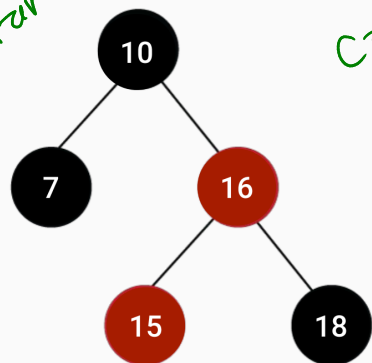
Inserta en hoja



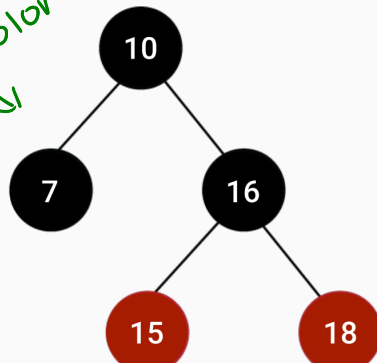
LR y LL

Rotas

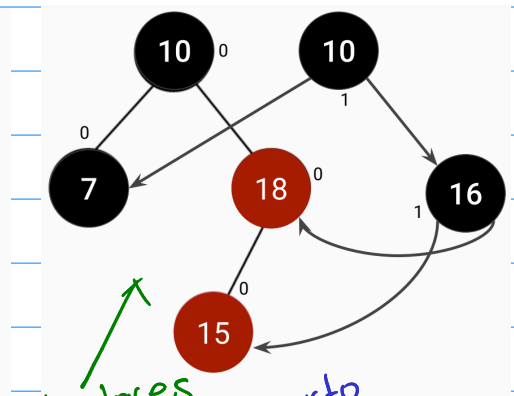
Resultado de rotar



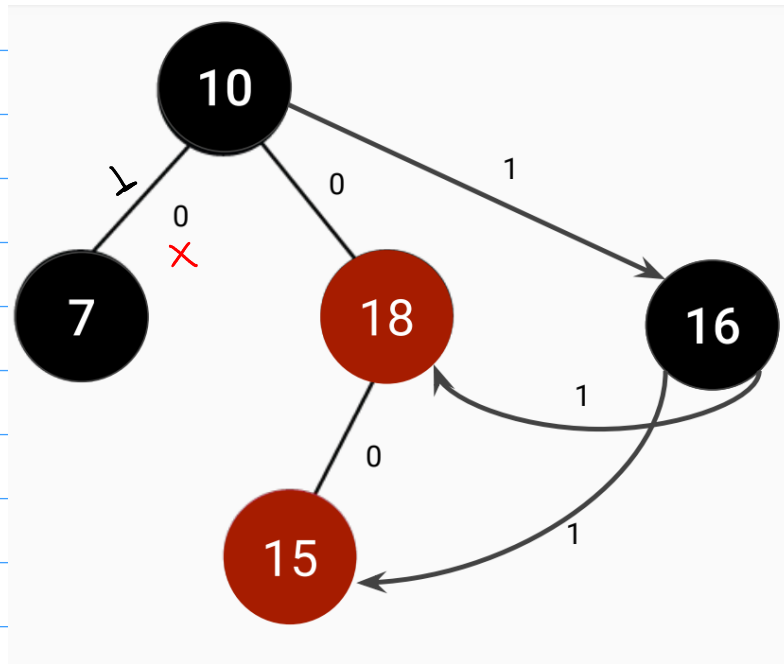
Cambio de color



Apuntadores
0 \rightarrow NODO muerto
L \rightarrow NODO vivo

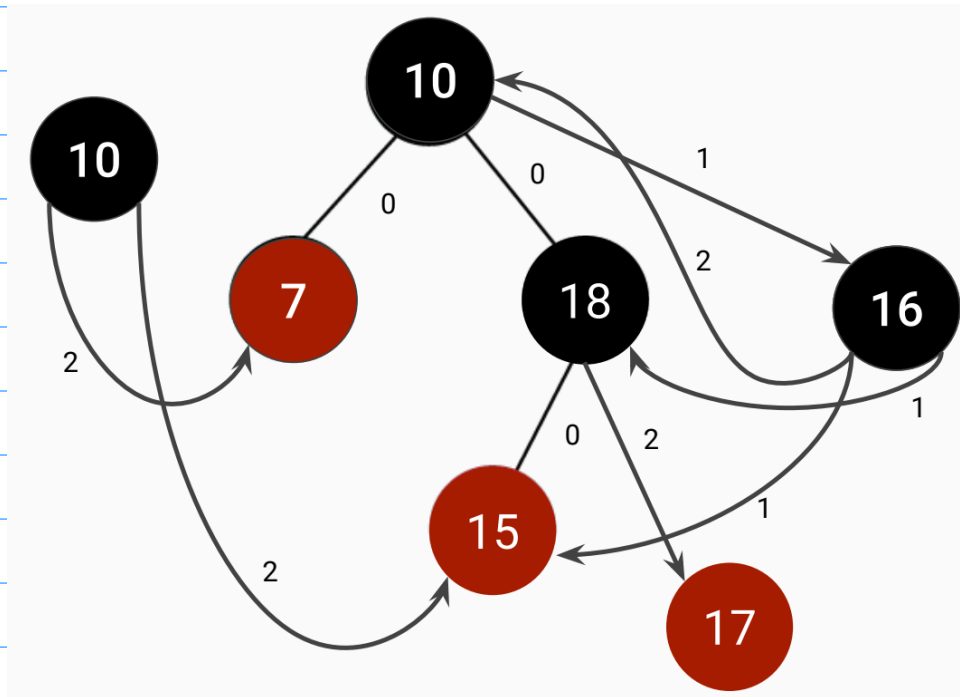


Reducimos el número de apuntadores.



Obs. Conforme se insertan más elementos el index de las aristas aumenta, entonces el árbol actual se forma de "recorrer" el árbol con aristas de index X donde $\{0, 1, \dots, X\}$ y X el número de inserciones o eliminaciones.

Resultado de insertar 17 →



NOTA. Realizar eliminaciones.