**Tipos de dato abstracto**

Cola con prioridad: la prioridad se puede manejar de cualquier forma (mayores a menores o viceversa)

Poner elementos: O(n)

Sacar elementos: O(1)

Si quiero que poner sea O(1) hago insertar al final, el problema de esto es que sacar se vuelva O(n).

**Montículos**: es un TDA, es un árbol binario casi completo (las hojas se van completando de izquierda a derecha) / completo. **El padre siempre tiene que tener más prioridad que los hijos y eso se debe dar en todos los subárboles**. No importa si los más grandes están a la derecha o izquierda. Solo importa la condición anterior.

El elemento que siempre sale es la raíz que es la que tiene más prioridad.

Se comienza por las hojas.

Poner elemento: si el nodo a insertar no cumple la condición (tiene más prioridad que el padre), se cambian de posición mutuamente.

**O(ln(n))**

Sacar elemento: se saca de la raíz que siempre tiene la prioridad más alta. Tomo el más grande de los menores y lo pongo en la raíz, comparo con sus hijos y lo cambio por el mayor, hago lo mismo hacia abajo hasta que quede en la posición correspondiente.

**O(ln(n))**

Para encontrar al padre de cualquier elemento del montículo (que lo aplicamos en un vector), se divide por dos al i (parte entera).

Para el hijo izquierdo hago i\*2

Para el hijo derecho hago (i\*2)+1

Se pueden usar montículos en cualquier caso donde pueda aplicar una cola.