

ESTRUCTURAS DE DATOS

dict_heap

En este modulo se incluye una versión de la estructura *cola prioritaria* implementada mediante un diccionario, que permite el acceso a los elementos en tiempo lineal.

Funciones

- **min_heap(iterable, set_max, key):** devuelve un objeto de tipo dict_heap. Si se proporciona un iterador lo genera a partir del iterador con coste lineal. Con **key** se puede definir una funcion de comparacion.
- **peek():** permite ver (sin eliminar) el elemento con la llave más pequeña del heap.
- **pop():** permite ver y eliminar el elemento con la llave más pequeña del heap.
- **insert(key, item):** permite insertar un elemento en el heap. Para usar el item como llave se puede usar el constructor con un elemento.

Costes de la operaciones

- **insertar:** coste $O(\log(n))$ promedio.
- **extraer:** coste $O(\log(n))$ promedio.
- **modificar prioridad:** coste $O(\log(n))$ promedio.
- **crear un heap:** coste $O(n)$.

x^2