

Comm:

Se usa la estructura hash para guardar las lineas de un archivo (en $O(1)$) y así luego poder ver si las lineas del otro ya se encuentran, o no, en el hash (en $O(1)$) y hacer lo que corresponda según el modo dado.

En modo 0 y 2 solo se imprime la linea. Pero en modo 1 si la linea del segundo archivo ya esta en el hash se la borra del hash (en $O(1)$) y una vez leído el segundo archivo se itera el hash imprimiendo todas las lineas (en $O(k)$, siendo k la cantidad de lineas).

Actualizar prioridad:

Una vez encontrado el elemento en el arreglo del heap ($O(n)$):

- Si padre < elemento: upheap ($O(\log n)$) al elemento ya que la invariante se rompe con el padre.
- Si algun hijo > elemento: downheap ($O(\log n)$) al elemento ya que la invariante se rompe con ese hijo.

iter_postorder externo:

Para iterador externo se utiliza una pila para ir des/apilando ($O(1)$) los elementos del arbol y así recorrerlos en postorder. Para obtener el elemento actual o ver si se esta al final tambien es $O(1)$.

Se utiliza una pila porque con algunas reglas para el des/apilado de los elementos al avanzar se puede recorrer al arbol sin inconvenientes.

iter_postorder interno:

Recorre el arbol en postorder de forma recursiva, ya que al tener que repetir el mismo proceso para cada nodo es más conveniente que hacerlo iterativamente.