



Multiestructuras

El todo es mayor que la suma de las partes



Intro

Muchas veces el uso exclusivo de una sola estructura no nos permite llegar a órdenes deseados.

Recurrir a más de una estructura puede solucionar dichas situaciones siempre y cuando se elijan y analisen correctamente.

Recurrentemente se plantean problemas para ser resueltos por una “Estructura” padre, que contiene estructuras “atómicas” que ya hemos visto antes (listas, árboles, tablas de hash, etc).



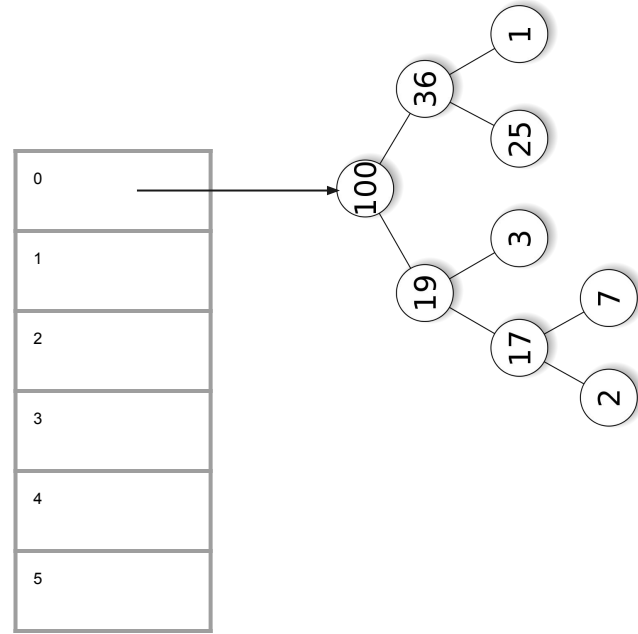
Se intenta usar dos o más estructuras para resolver los problemas

	insertar	eliminar	obtener min/max	eliminar min/max
Lista	$O(1)$	$O(N)$	$O(N)$	$O(N)$
Lista Ord	$O(N)$	$O(N)$	$O(1)$	$O(1)$
ABB	$O(Ln)$ cp / $O(N)$ pc	$O(Ln)$ cp / $O(N)$ pc	$O(Ln)$ cp / $O(N)$ pc	$O(Ln)$ cp / $O(N)$ pc
AVL	$O(Ln)$	$O(Ln)$	$O(Ln)$	$O(Ln)$
Heap	$O(1)$ cp / $O(Ln)$ pc	$O(N)$	$O(1)$	$O(Ln)$
Tabla de hash	$O(1)$ cp / $O(N)$ pc	$O(1)$ cp / $O(N)$ pc	-	-

A tener siempre en cuenta cuando resolvemos estos ejercicios

Y no debemos limitarnos sólo a lo que hemos visto

Por ejemplo, las tablas de hash abiertas no tienen porque ser usadas con listas. ¿Qué tal si usamos un AVL? ¿Cómo afectan los órdenes?



THINK
OUTSIDE
THE BOX

X	O	O
O	O	X
X	X	O

Ejemplo: Ranking FIFA

Ejemplo Ranking FIFA

Se desea implementar un sistema donde se almacene y manipule el ranking FIFA, es decir, el ranking de fútbol a nivel de país.

Cada vez que se ingresa un nuevo país, este debe ir último en el ranking.

La única forma de subir en el ranking es retando quien está por encima de la posición actual. Por ejemplo, el 4to puesto puede retar al 3ero, el 9no al 8vo, etc. De ganar el reto, intercambian posiciones.



Ejemplo Ranking FIFA: funciones

- // pre: no existe el país dentro del ranking
// pos: se agrega el país en la última posición del ranking
void **agregarPais**(string nombrePais)
- // pre: el país existe dentro del ranking
// pos: retorna la posición actual del país dentro del ranking
int **posicionRanking**(string nombrePais)
- // pre: que existe esa posición dentro del ranking
// pos: retorna el país en dicha posición
string **posicionPais**(int unaPosicion)
- // pre: el país existe dentro del ranking y no está en el primer puesto
// pos: de ganar el país retador intercambia posiciones con el país retado
void **retar**(string paisRetador, bool ganoRetador)



Discutamos

void **agregarPais**(string nombrePais)

int **posicionRanking**(string nombrePais)

string **posicionPais**(int unaPosicion)

void **retar**(string paisRetador, bool ganoRetador)



Solución 1: usar un array

```
void agregarPais(string nombrePais)
```

```
int posicionRanking(string nombrePais)
```

```
string posicionPais(int unaPosicion)
```

```
void retar(string paisRetador, bool ganoRetador)
```

0	-
1	Bélgica
2	Francia
3	Brasil
4	Inglaterra
5	Uruguay
...	...



Solución 1: usar un array

void agregarPais(string nombrePais) $O(1)$

int posicionRanking(string nombrePais) $O(N)$

string posicionPais(int unaPosicion) $O(1)$

void retar(string paisRetador, bool ganoRetador) $O(N)$

0	-
1	Bélgica
2	Francia
3	Brasil
4	Inglaterra
5	Uruguay
...	...



T <-> int

—



T \leftrightarrow int

Consiste en usar una tabla de hash<T, int> y un array de T (T[]).

La tabla de hash hace la función de **T -> int**, mientras que el array **int -> T**.

Nota: Puede ser útil en otros contextos del curso, por ejemplo en Heap o Grafos.

Solución 2: array + tabla de hash

```
void agregarPais(string nombrePais)
```

```
int posicionRanking(string nombrePais)
```

```
string posicionPais(int unaPosicion)
```

```
void retar(string paisRetador, bool ganoRetador)
```

string (país)	int (posicion)
Francia	2
Uruguay	5
Bélgica	1
Croacia	6
....

0	-
1	Bélgica
2	Francia
3	Brasil
4	Inglaterra
5	Uruguay
...	...



Solución 2: array + tabla de hash

void agregarPais(string nombrePais) $O(1)$ cp

int posicionRanking(string nombrePais) $O(1)$ cp

string posicionPais(int unaPosicion) $O(1)$ pc

void retar(string paisRetador, bool ganoRetador) $O(1)$ cp

string (país)	int (posicion)
Francia	2
Uruguay	5
Bélgica	1
Croacia	6
....

0	-
1	Bélgica
2	Francia
3	Brasil
4	Inglaterra
5	Uruguay
...	...



Ejemplo: parcial 10/19 noct

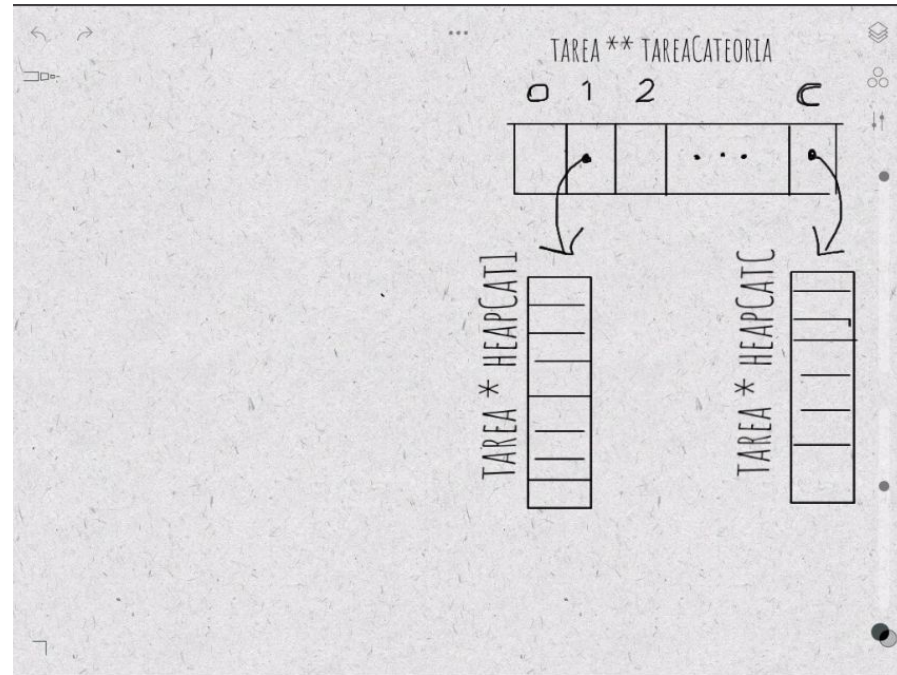
https://aulas.ort.edu.uy/pluginfile.php/59765/mod_folder/content/0/EstDatAlg2%201910pn.doc?forcedownload=1

Ejemplo: parcial 10/19 noct

Para guardar las tareas se decide tener C heaps.
Cada heap representa las tareas de una categoría X.

De ese modo:

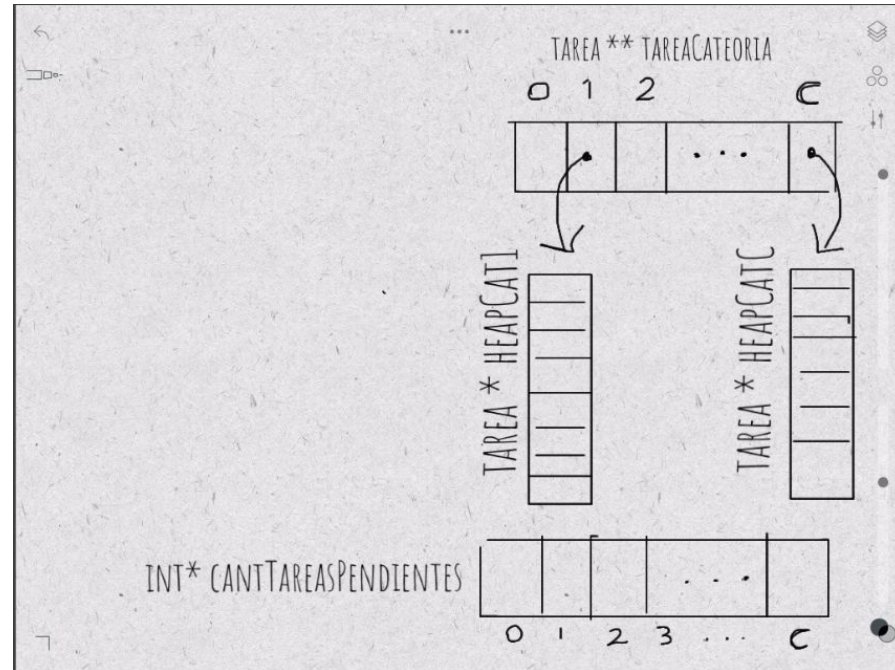
- 1) *Agregar una tarea*: consiste en agregar la tarea al heap correspondiente (depende de la cat.)
- 2) *Ver la tarea más prioritaria de una categoría*: se utiliza la función **top** la cual retorna el elemento con mayor prioridad pero no lo elimina.
- 3) *Resolver tarea más prioritaria de una categoría*: utiliza la función **pop** del heap correspondiente



Ejemplo: parcial 10/19 noct

Para contar las tareas pendientes de categorías se tiene un array de int de tamaño $C+1$.

- 1) *Agregar una tarea*: suma 1 al casillero correspondiente.
- 3) *Resolver tarea más prioritaria de una categoría*: resta en 1 al casillero correspondiente.
- 4) *Cantidad de tareas*: consulta el array de cantidades en la posición que corresponde a la categoría.

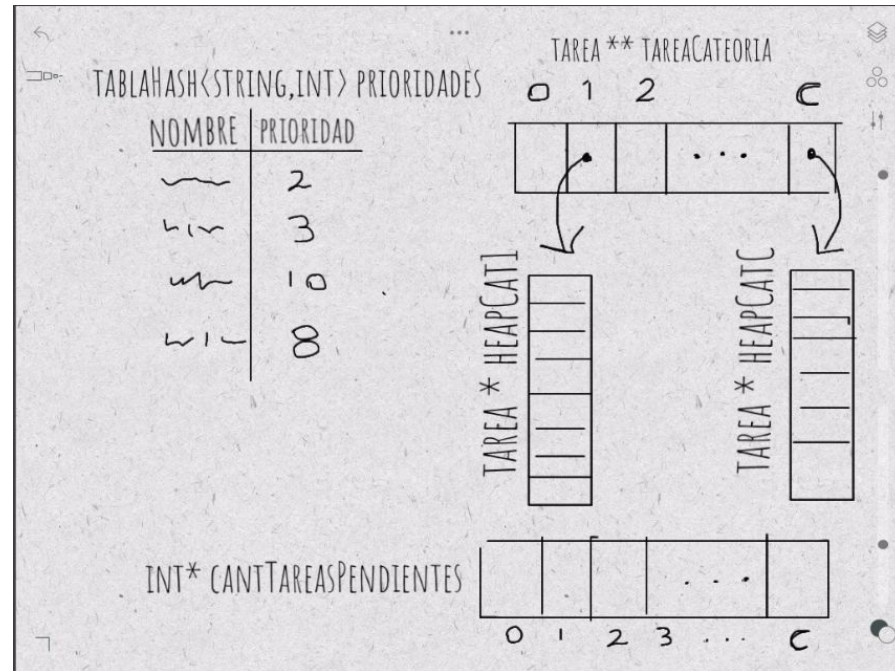


Ejemplo: parcial 10/19 noct

Por último, tenemos una tabla de hash, con clave string (nombre de la tarea) y valor int (prioridad de la tarea).

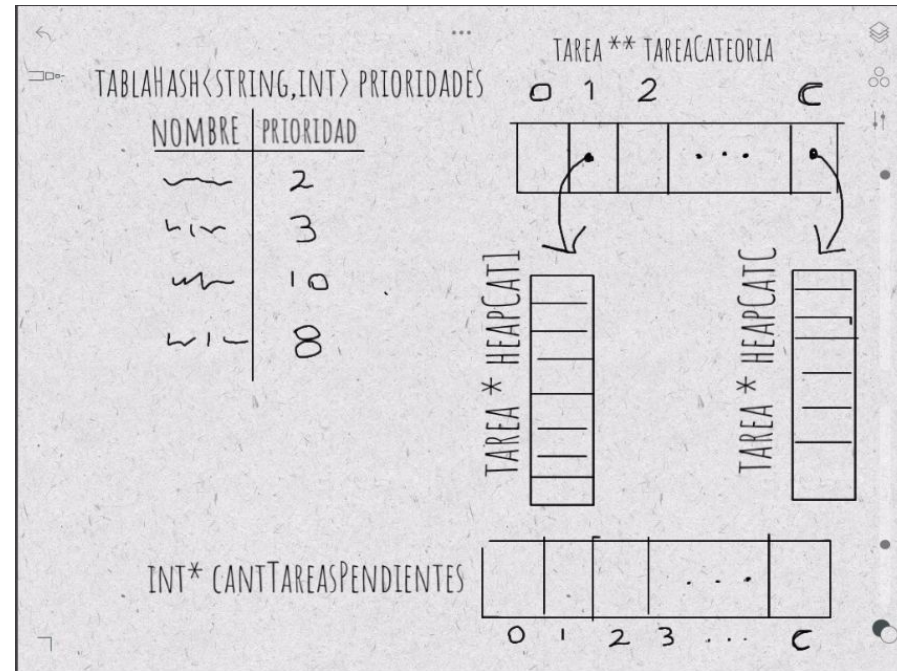
1) *Agregar una tarea*: ahora también agregamos el nombre de la tarea con su prioridad la tabla de hash.

5) *Obtener prioridad de una tarea*: esto simplemente consiste en recuperar el valor a partir del nombre de la tarea en la tabla de hash.



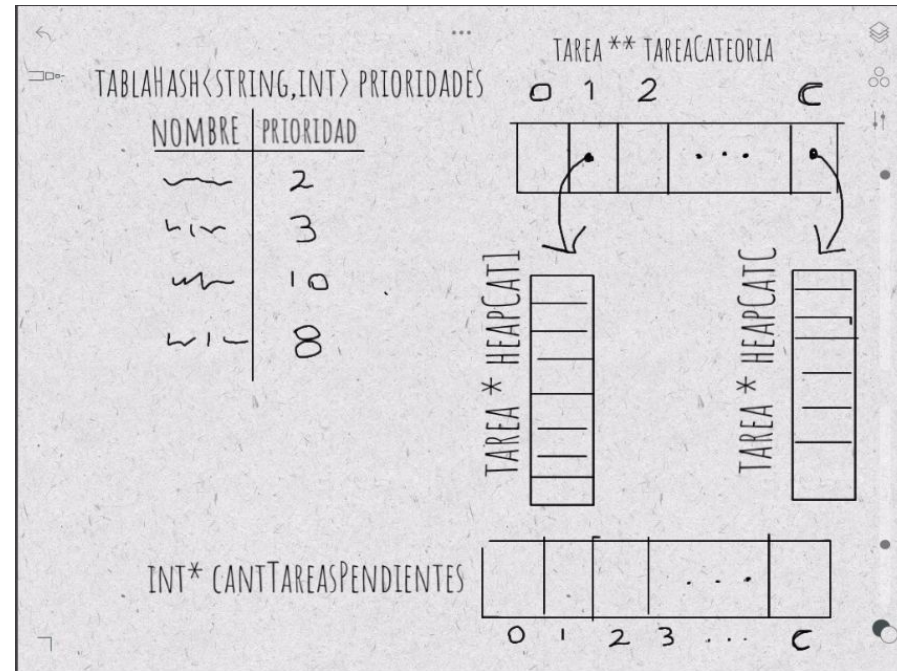
Ejemplo: parcial 10/19 noct

Agregar una tarea	$O(\lg N)_{pc}$
Ver la tarea más prioritaria de una categoría	
Resolver tarea más prioritaria de una categoría	
Cantidad de tareas	
Obtener prioridad de una tarea	



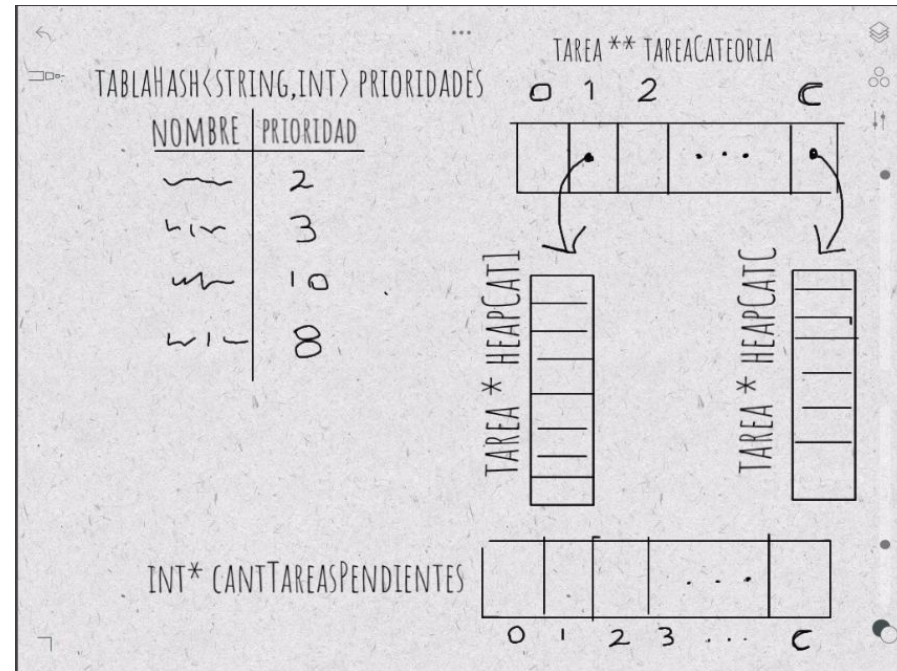
Ejemplo: parcial 10/19 noct

Agregar una tarea	$O(\lg N) + O(1)_{pc}$
Ver la tarea más prioritaria de una categoría	
Resolver tarea más prioritaria de una categoría	
Cantidad de tareas	
Obtener prioridad de una tarea	



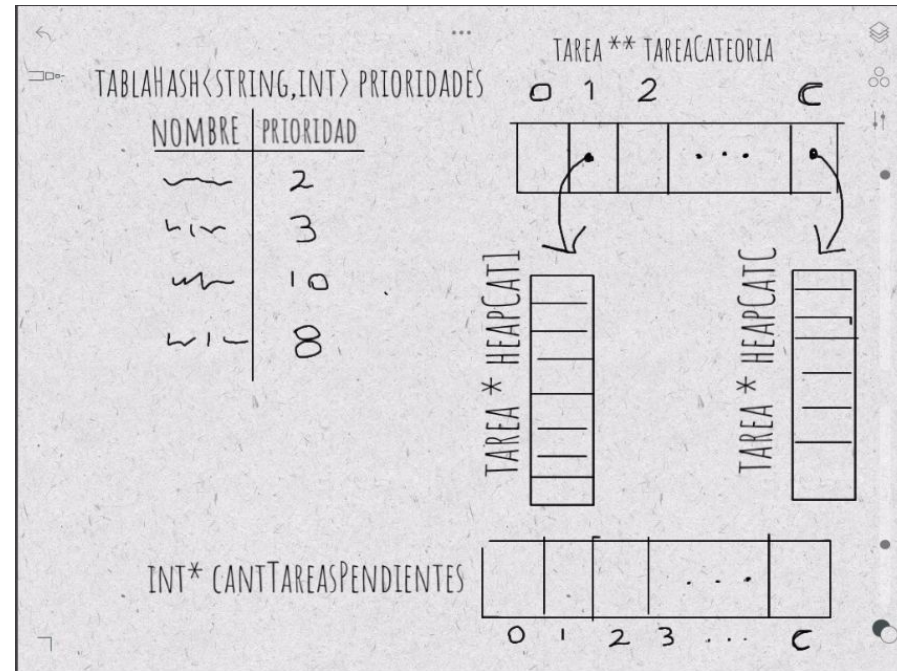
Ejemplo: parcial 10/19 noct

Agregar una tarea	$O(\lg N) + O(1) + O(N)_{pc}$
Ver la tarea más prioritaria de una categoría	
Resolver tarea más prioritaria de una categoría	
Cantidad de tareas	
Obtener prioridad de una tarea	



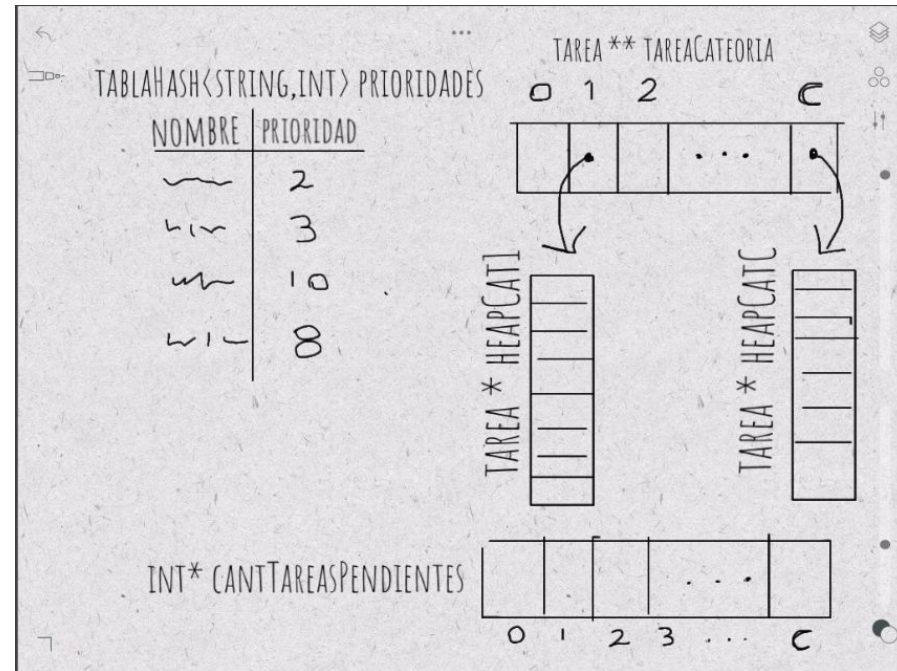
Ejemplo: parcial 10/19 noct

Agregar una tarea	$O(\lg N) + O(1) + O(1)_{pc}$
Ver la tarea más prioritaria de una categoría	
Resolver tarea más prioritaria de una categoría	
Cantidad de tareas	
Obtener prioridad de una tarea	



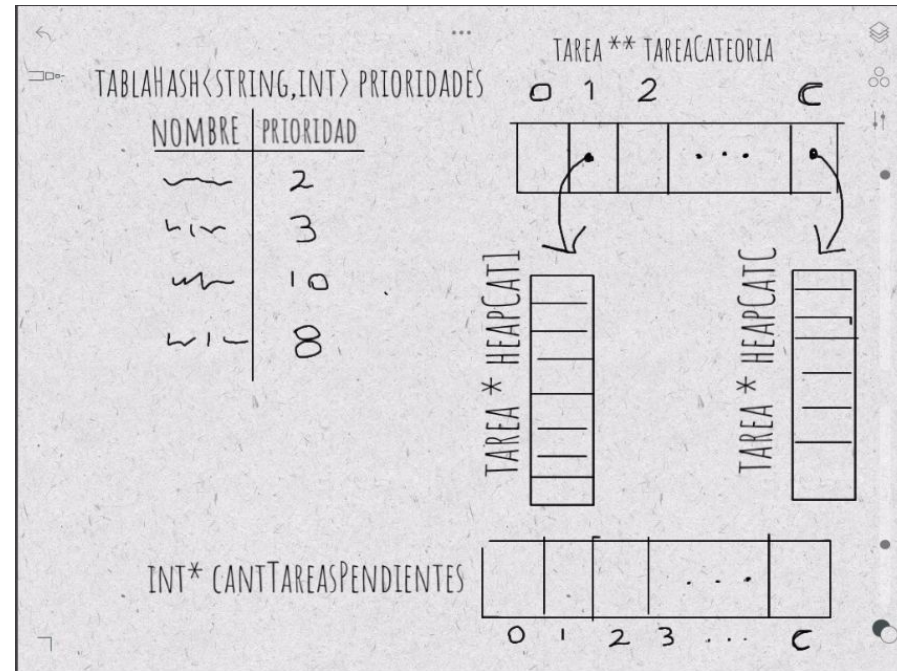
Ejemplo: parcial 10/19 noct

Agregar una tarea	$O(\lg N)_{pc}$
Ver la tarea más prioritaria de una categoría	$O(1)_{pc}$
Resolver tarea más prioritaria de una categoría	
Cantidad de tareas	
Obtener prioridad de una tarea	



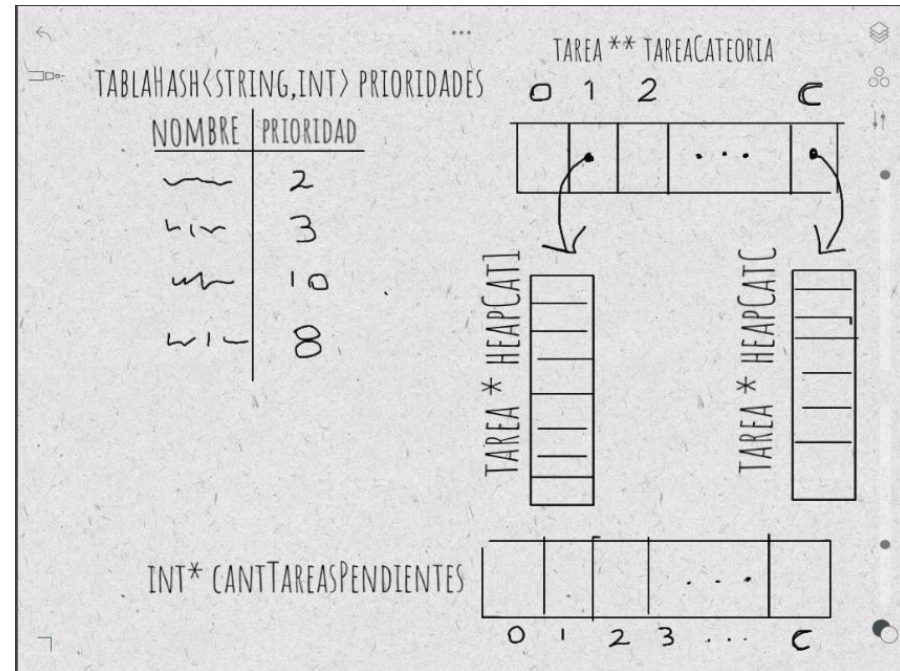
Ejemplo: parcial 10/19 noct

Agregar una tarea	$O(\lg N)_{pc}$
Ver la tarea más prioritaria de una categoría	$O(1)_{pc}$
Resolver tarea más prioritaria de una categoría	$O(\lg N) + O(1)_{pc}$
Cantidad de tareas	
Obtener prioridad de una tarea	



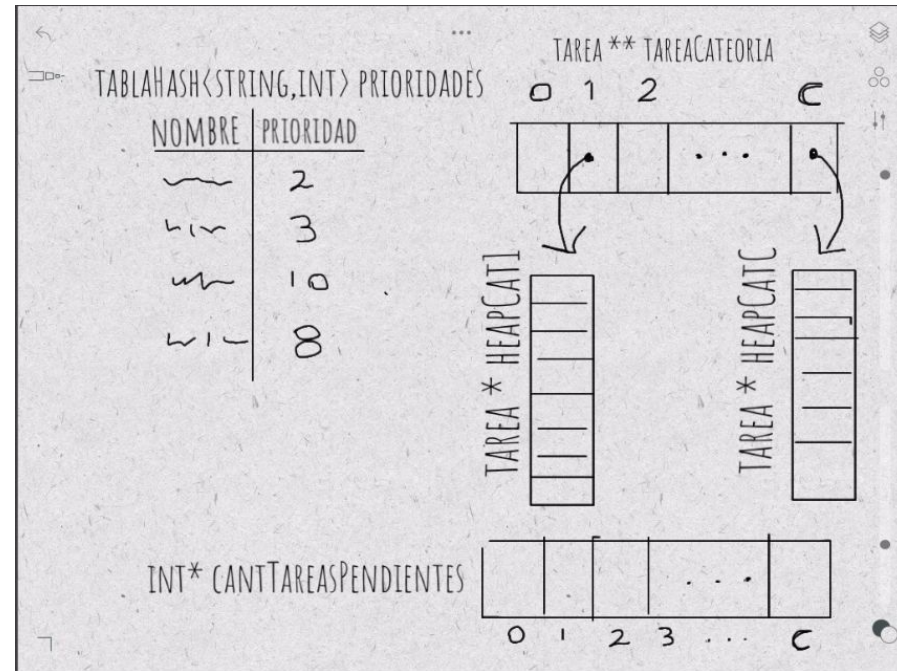
Ejemplo: parcial 10/19 noct

Agregar una tarea	$O(\lg N)_{pc}$
Ver la tarea más prioritaria de una categoría	$O(1)_{pc}$
Resolver tarea más prioritaria de una categoría	$O(\lg N)_{pc}$
Cantidad de tareas	$O(1)_{pc}$
Obtener prioridad de una tarea	



Ejemplo: parcial 10/19 noct

Agregar una tarea	$O(\lg N)_{pc}$
Ver la tarea más prioritaria de una categoría	$O(1)_{pc}$
Resolver tarea más prioritaria de una categoría	$O(\lg N)_{pc}$
Cantidad de tareas	$O(1)_{pc}$
Obtener prioridad de una tarea	$O(1)_{cp}$



Ejemplo: parcial 10/19 mat

https://aulas.ort.edu.uy/pluginfile.php/59765/mod_folder/content/0/EstDatAlg2%201910pm.doc?forcedownload=1

Ejemplo: parcial 05/19 noct

https://aulas.ort.edu.uy/pluginfile.php/59765/mod_folder/content/0/EstDatyAlg2%201905pn.pdf?forcedownload=1

Ejemplo: parcial 05/18 noct

https://aulas.ort.edu.uy/pluginfile.php/59765/mod_folder/content/0/EstDatyAlg2%201805%20pn.pdf?forcedownload=1