

## Auxiliar 8 - Colas de Prioridad y Búsqueda Secuencial

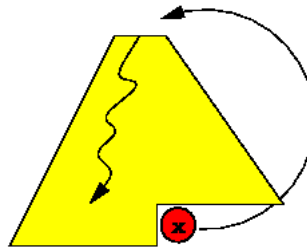
Profesores: Nelson Baloian, Patricio Poblete  
Benjamín Bustos  
Sebastian Ferrada

Auxiliares: Vicente Olivares, Ricardo Valdivia  
Sebastián Acuña  
Valentina Aravena

### P1. Heap

En clase se vio un algoritmo para extraer el máximo elemento de un **Heap** que consiste en:

- 1 Remover el nodo de la raíz
- 2 Reemplazar la raíz por el último elemento
- 3 Hundir el nodo hasta que se vuelva a cumplir el orden de heap



Ahora, utilizando la clase `Heap` del apunte:

- (a) Implemente la función `hundir_menos_infinito(a, j, n)`, que sin importar el valor original de `a[j]` deje su nodo en el último nivel, y retorne la nueva posición.
- (b) Utilizando la función anterior, implemente el método `extraer_max2(self)` como sustituto del algoritmo de extracción original.

### P2. Los $k$ menores

Dado un arreglo de  $n$  números enteros, diseñe un algoritmo que encuentre los  $k$  valores menores del arreglo en tiempo  $O(n \log k)$  en el peor caso y que utilice solo  $O(k)$  memoria adicional.

### P3. Extraer máximo modificado

Suponga que para extraer el máximo elemento de un *heap*, en vez de tomar el último elemento del arreglo y trasladarlo al lugar vacante, toma al mayor de los hijos de la raíz para rellenar dicho lugar. Diseñe un algoritmo que utilice esta estrategia para la extracción del máximo, considerando que al final del proceso de extracción el arreglo debe volver a cumplir con la propiedad de ser un *heap*.