

ÁRBOLES BINARIOS PARCIALMENTE ORDENADOS

Joaquín Fernández-Valdivia

Javier Abad

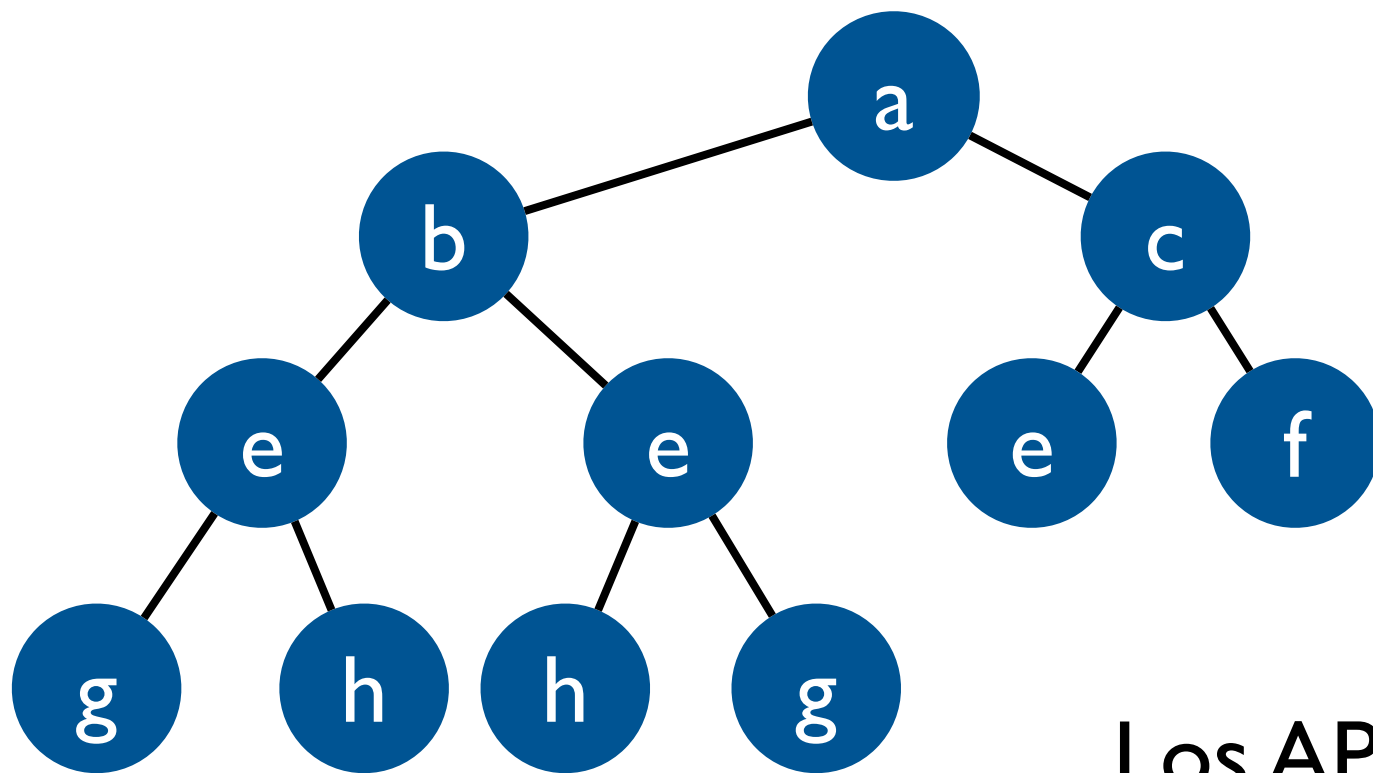
Dpto. de Ciencias de la Computación e Inteligencia Artificial

Universidad de Granada



Definición de APO

- Se dice que un árbol binario es un APO si cumple la condición de que la etiqueta de cada nodo es menor o igual que las etiquetas de los hijos, manteniéndose tan equilibrado (balanceado) como sea posible (hojas empujadas a la izquierda)



Sólo nos interesan funciones para:

- insertar elementos
- borrar la raíz
- consultar la raíz

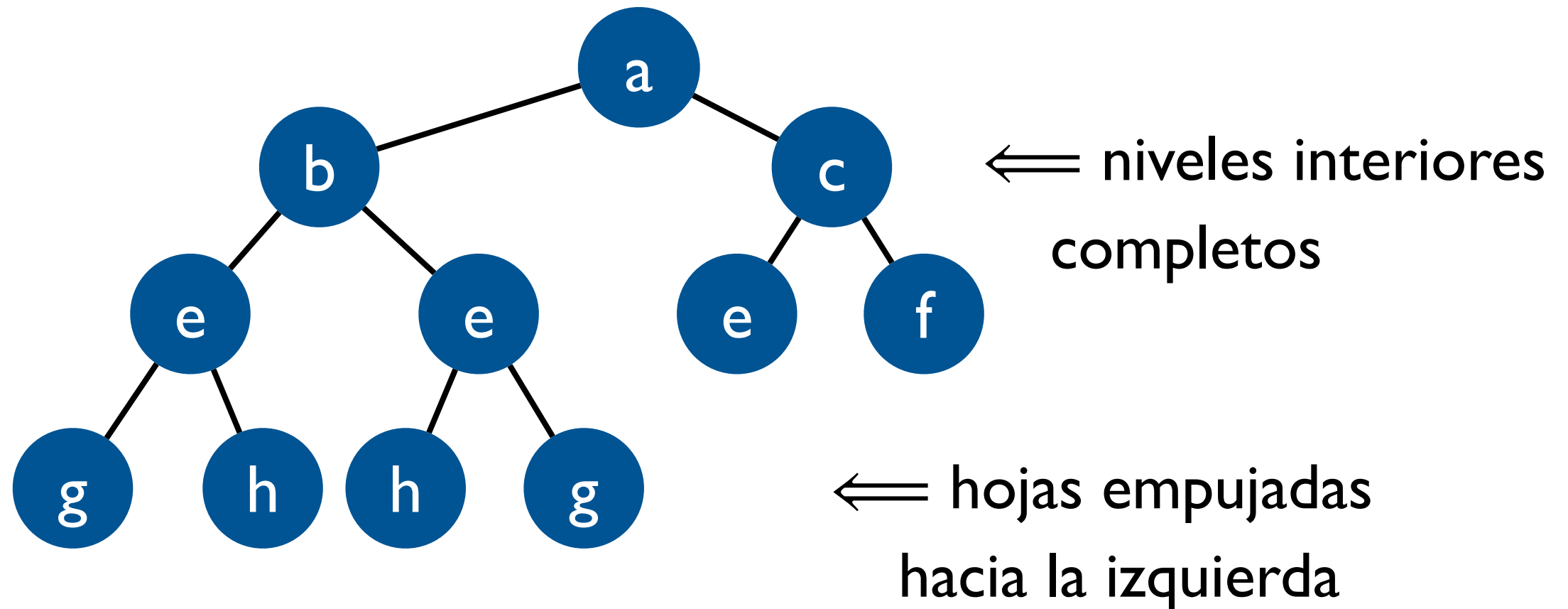
Los APO son útiles para ordenación
HEAPSORT

Representación: el montón

- La representación que usaremos para los APO es la del montón (Heap)
- Un montón, M , en este contexto, será un vector en el que guardaremos el APO por niveles, de forma que si existen n nodos:
 - $M[0]$ alojará a la raíz
 - Los hijos izquierdo y derecho (si existen) del nodo $M[k]$ estarán en $M[2k+1]$ y $M[2k+2]$, lo que equivale a decir que el padre de $M[k]$ es $M[(k-1)/2]$, $\forall k > 0$

Representación: el montón

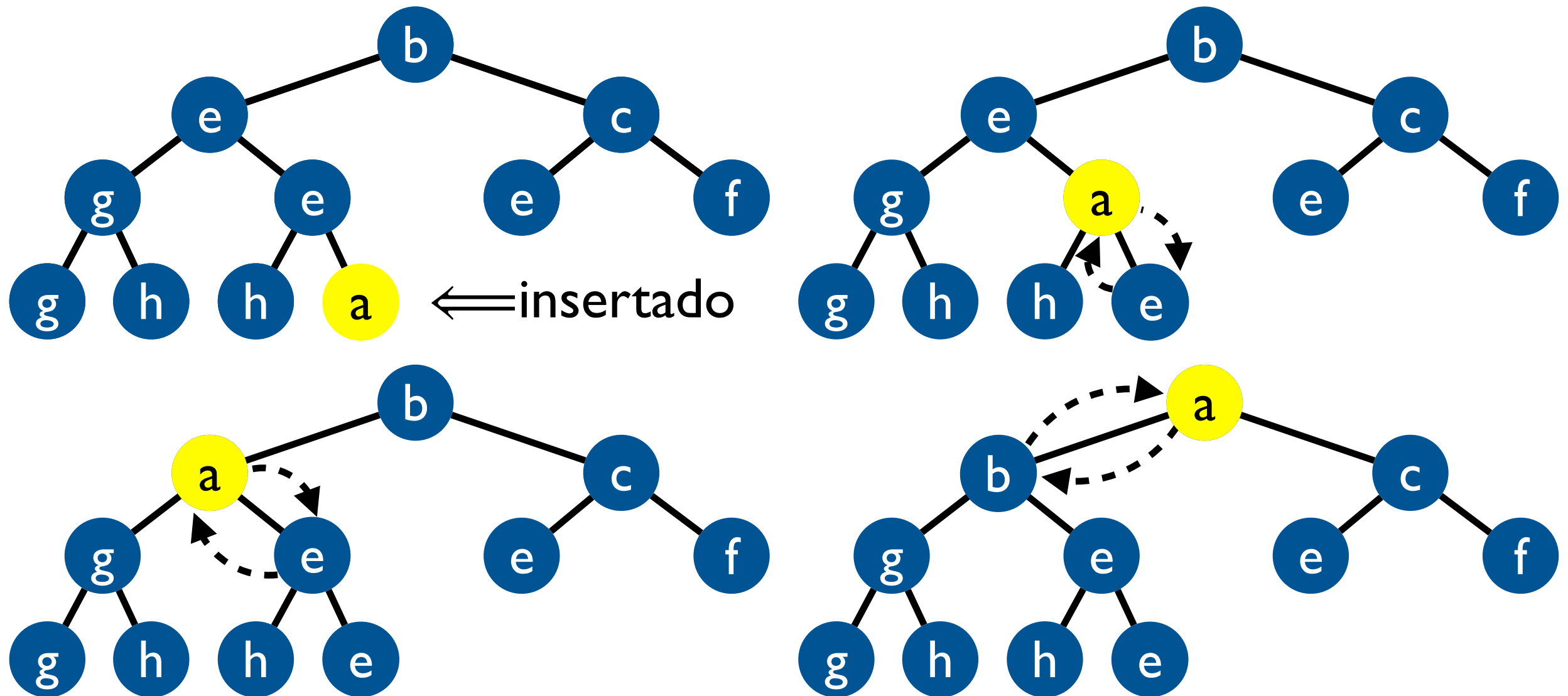
- El APO



se representa en el vector

0	1	2	3	4	5	6	7	8	9	10	...
a	b	c	e	e	e	f	g	h	h	g	...

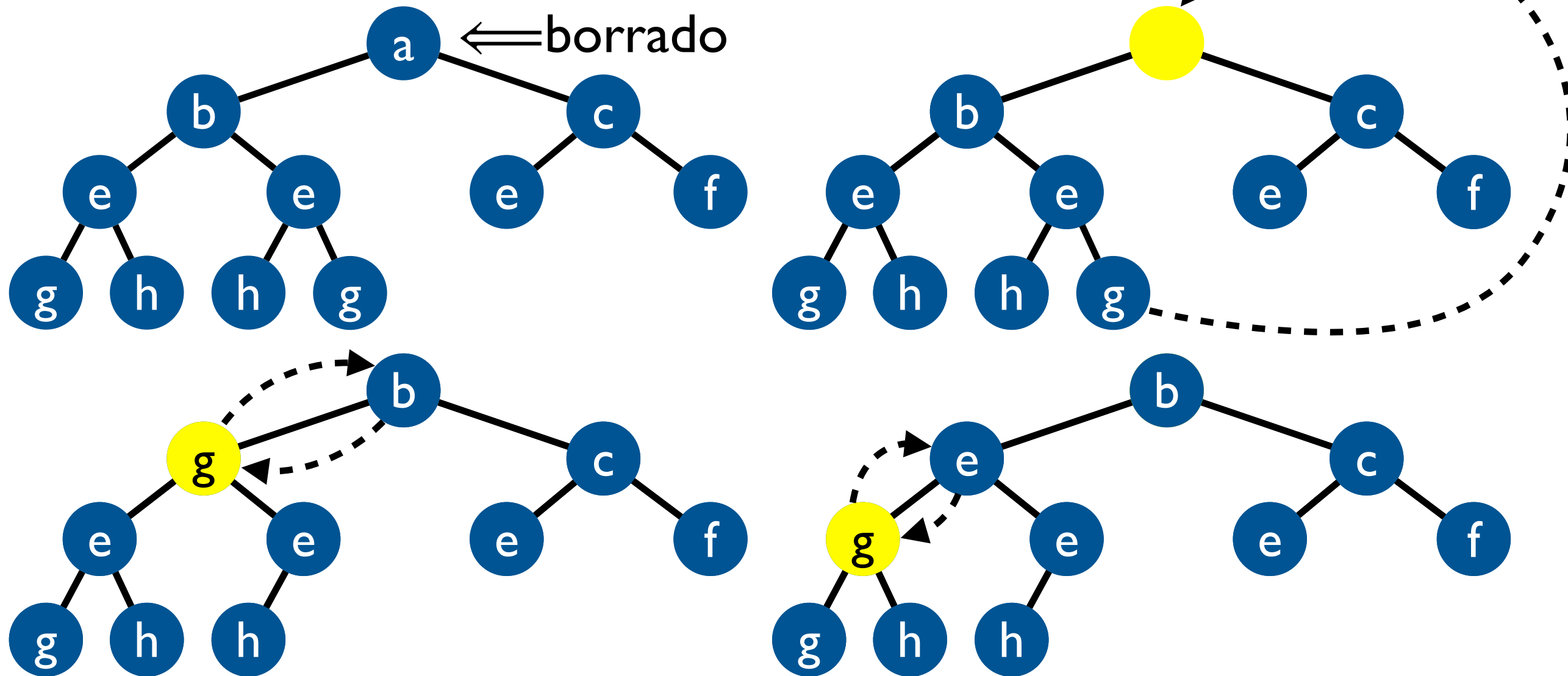
Inserción en el APO



$O(\log_2 n)$

Estas operaciones pueden realizarse gracias a la idea de mantener las hojas del APO empujadas a la izquierda

Borrado en el APO



$O(\log_2 n)$

Estas operaciones pueden realizarse gracias a la idea de mantener las hojas del APO empujadas a la izquierda

Consideraciones

- Si inserciones y borrados tienen $O(\log_2 n)$, ¿utilizar un APO para ordenar es mejor o peor que el algoritmo Quicksort? Con ambos tendríamos $O(n \log_2 n)$...
 - Con Quicksort (y con cualquier método habitual de ordenación), tenemos que reordenar si añadimos nuevos datos (o insertar ordenado).
 - Con Heapsort (usando un APO), cada inserción tiene sólo un coste de $O(\log_2 n)$ y los datos estarán "ordenados" (mejor dicho, preparados para obtenerse de forma ordenada).
- Heapsort es particularmente útil si el conjunto de datos es muy dinámico (inserciones y/o borrados)
- ¿Qué otra estructura de datos podemos implementar con un APO? (Pista: la habéis estudiado/usado este curso)

Ejercicio propuesto

- Sí, lo habéis adivinado (o no): podemos usar un APO para implementar una cola con prioridad.
- Recuerda, las operaciones que debe ofrecer una cola con prioridad son:
 - Frente: devuelve el elemento del frente
 - Poner: añade un elemento con la prioridad asociada
 - Quitar: elimina el elemento del frente
 - Vacía: indica si la cola está vacía

además de los métodos básicos (constructores por defecto y de copia, destructor y operador de asignación).

- Podemos obviar la operación Prioridad_Frente si seguimos la filosofía de la clase `priority_queue` de la STL: dejar en manos del usuario de la clase la definición del tipo base, al que sólo se le exige que tenga definido el operador `<`