

Montículos

Árboles de Decisión

Estructuras de Datos

Andrea Rueda

Pontificia Universidad Javeriana
Departamento de Ingeniería de Sistemas

Montículos

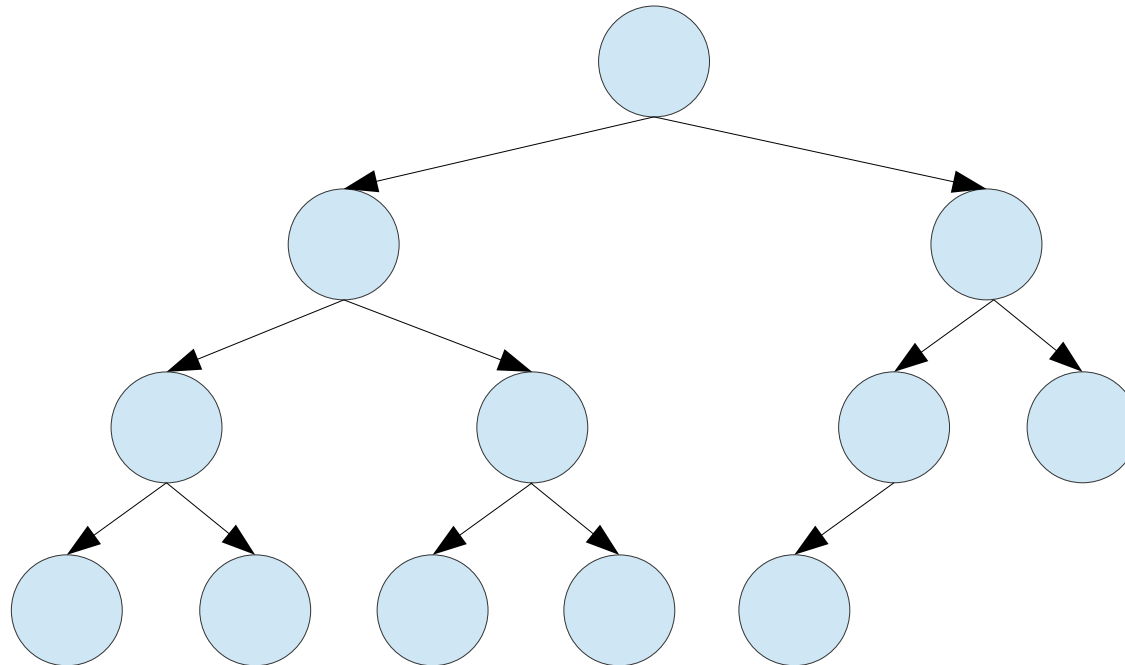
(*heaps*)

Montículo (*heap*)

- Estructura basada en árboles completos:

Todos los niveles, salvo el último, tienen el máximo número de nodos posibles.

En el último nivel, si no está lleno, todos los nodos deben estar concentrados hacia el lado izquierdo.



Montículo (*heap*)

- Satisface la propiedad:

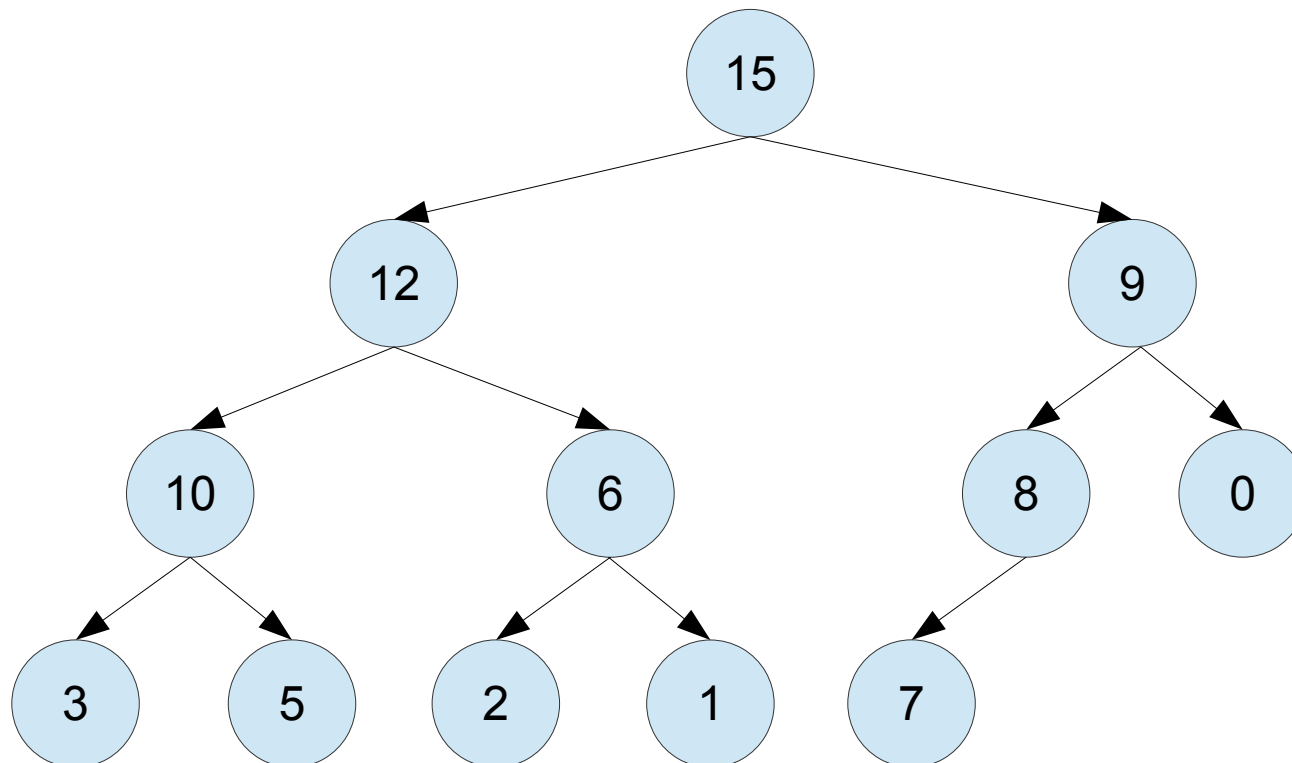
Si A es el nodo padre de B, el valor de A está ordenado con respecto al valor de B con la misma propiedad de ordenamiento a lo largo del montículo.

- Montículo máximo (*max heap*): si el valor del padre es siempre mayor o igual que el valor de sus hijos (máximo en la raíz).
- Montículo mínimo (*min heap*): si el valor del padre es siempre menor o igual que el valor de sus hijos (mínimo en la raíz).

Montículo (*heap*)

- No importa la relación de orden entre hermanos.

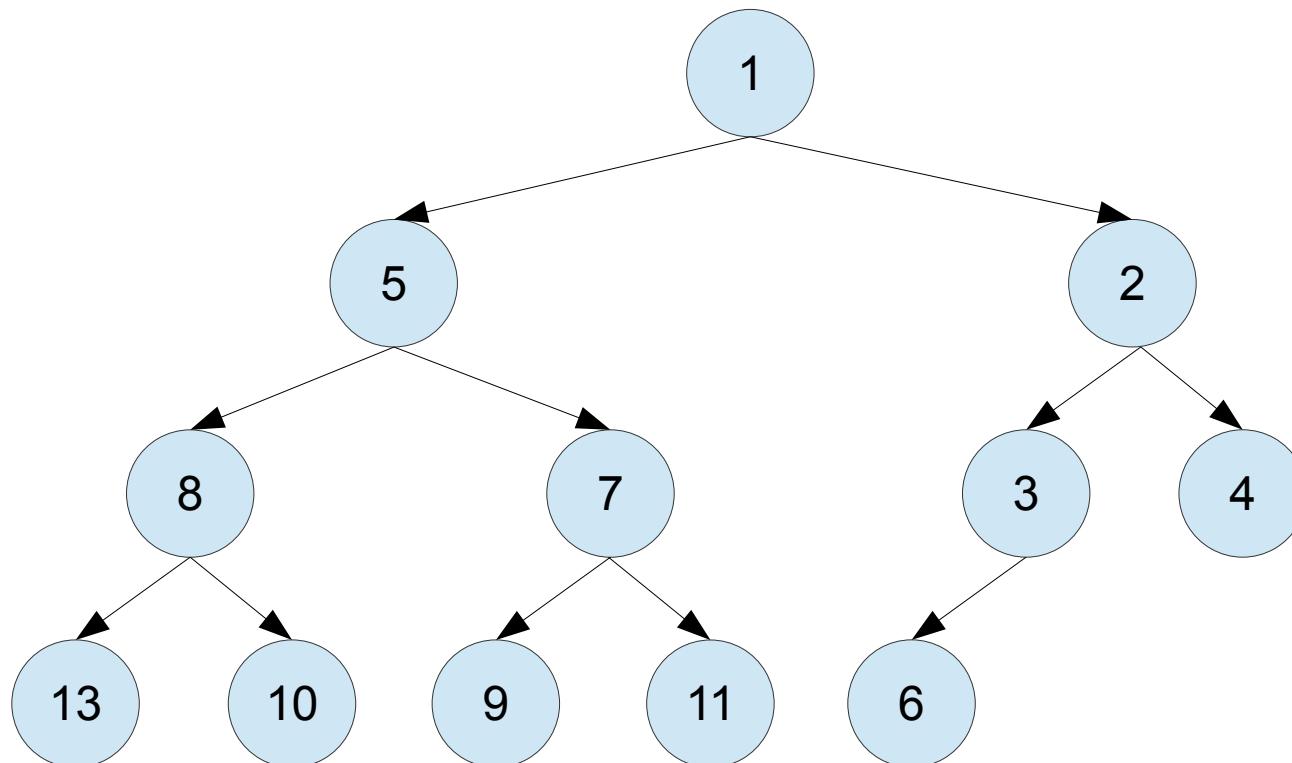
Ejemplo *max heap*:



Montículo (*heap*)

- No importa la relación de orden entre hermanos.

Ejemplo *min heap*:



Montículo (*heap*)

- Inserción:

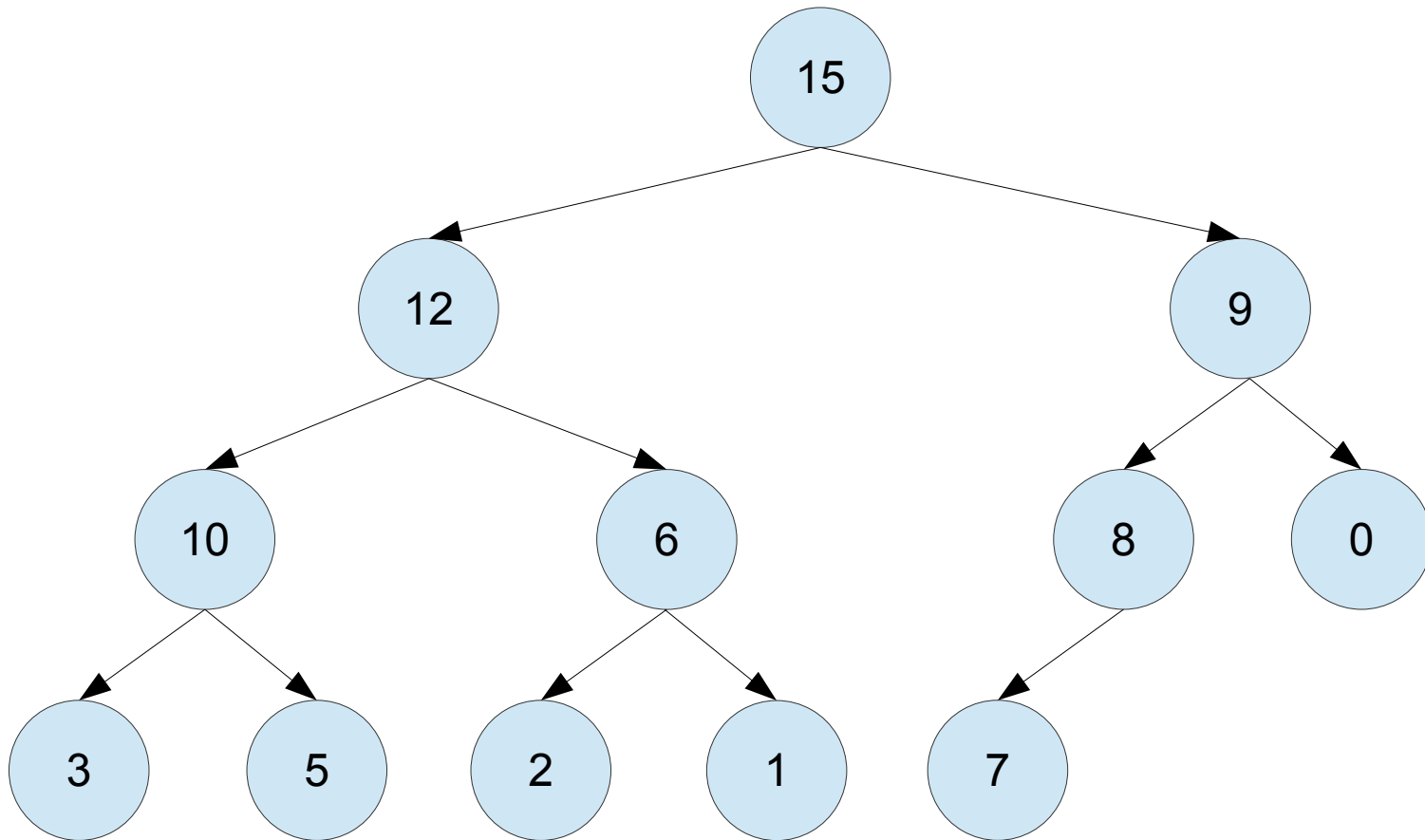
Puede requerir el restablecimiento de la propiedad de ordenamiento del montículo.

1. Añadir el elemento en la “última” posición.
2. Comparar el valor del nodo con su padre:
 - 2.1 si están en el orden correcto, terminar.
 - 2.2 si no, intercambiar el elemento con su padre y retornar al paso 2.

Depende de la altura del montículo $\rightarrow O(\log n)$.

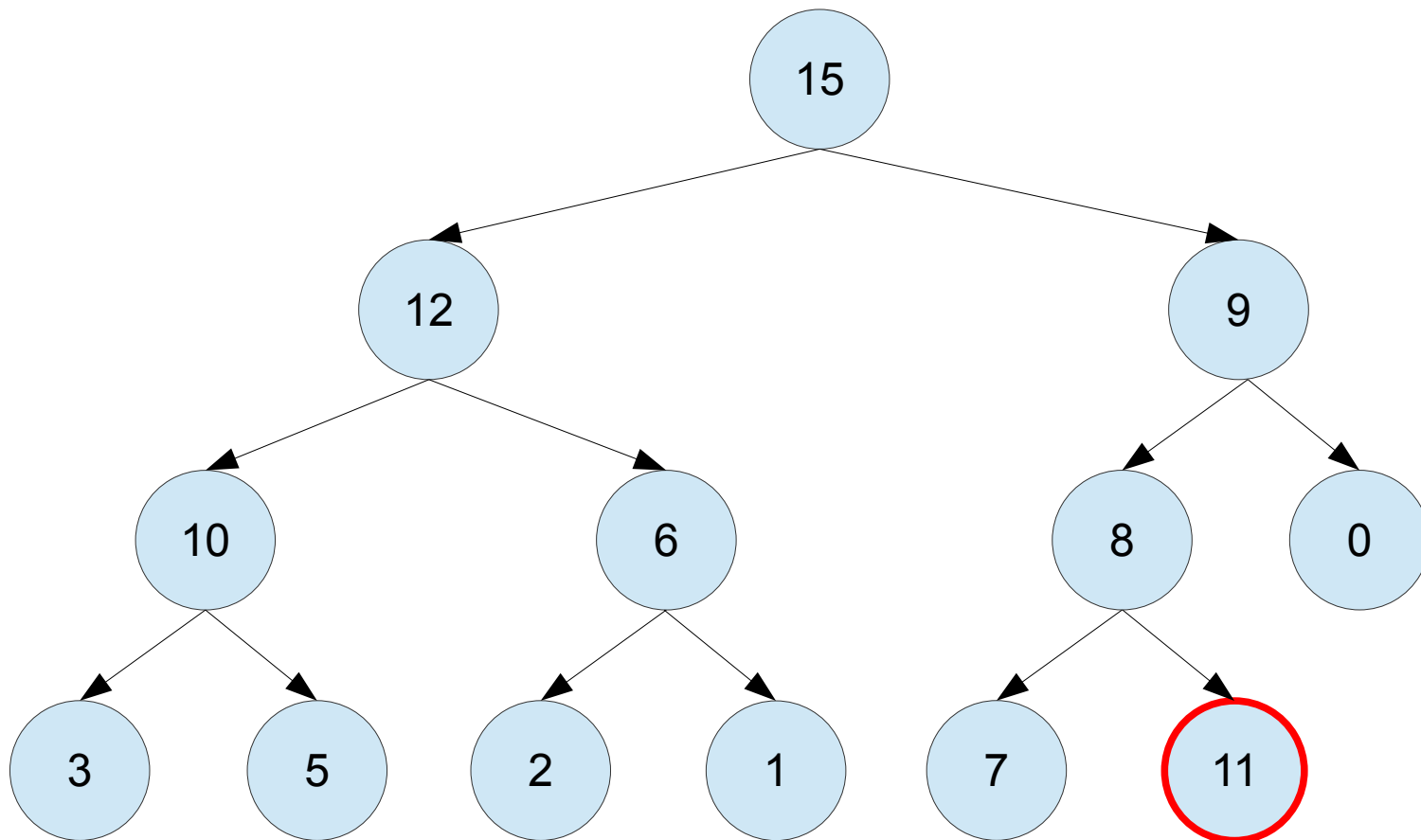
Montículo (*heap*)

- Inserción: insertar 11.



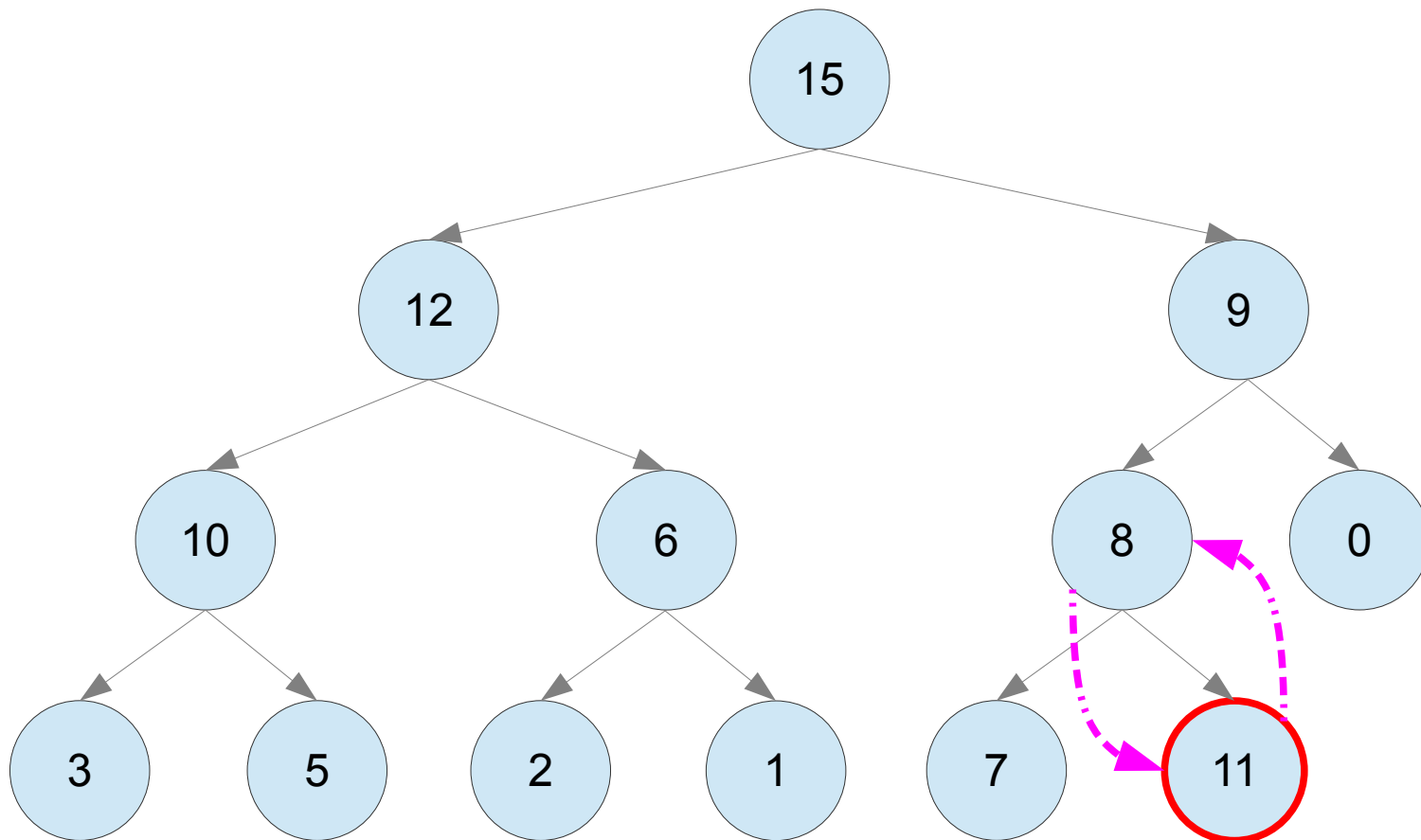
Montículo (*heap*)

- Inserción: insertar 11.



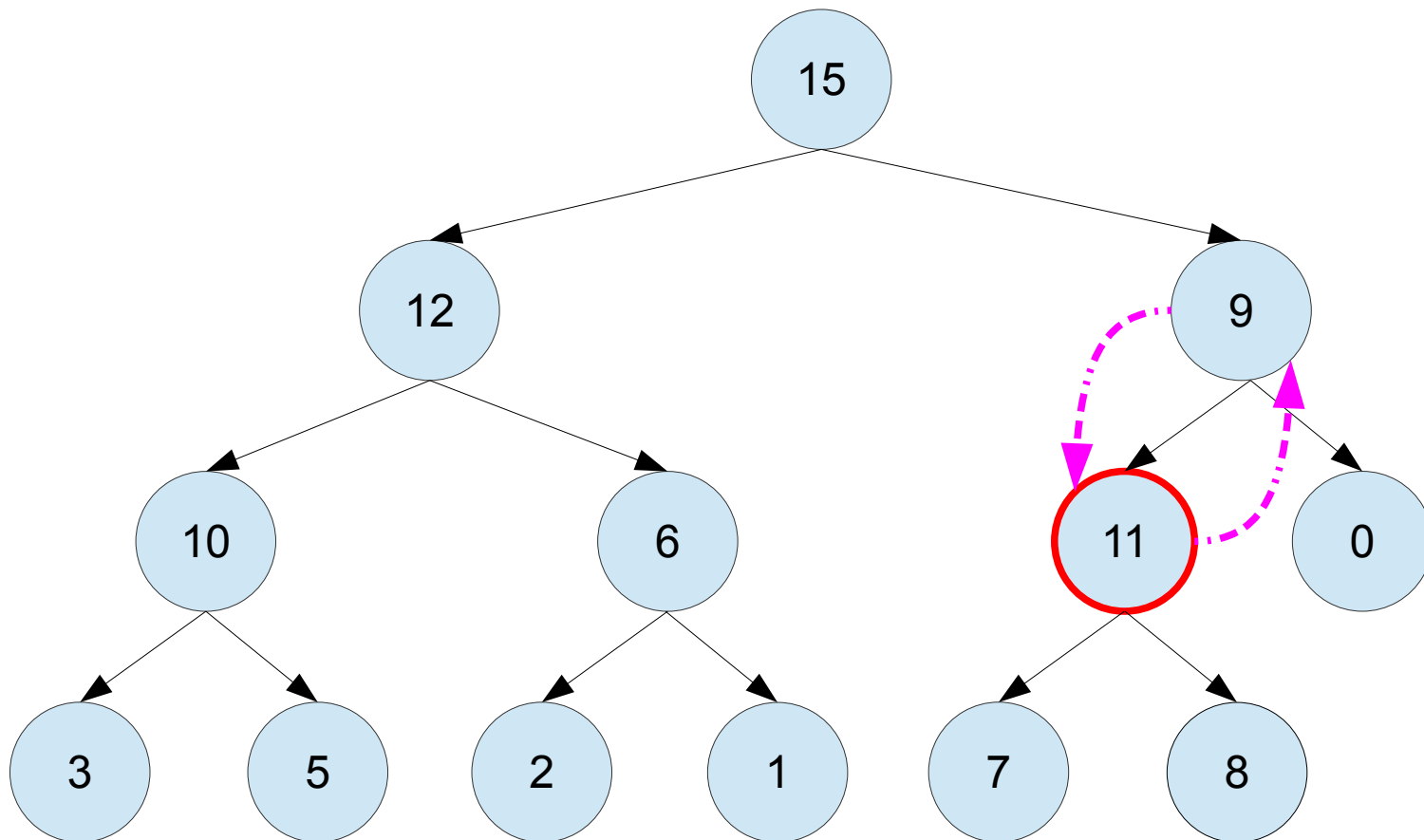
Montículo (*heap*)

- Inserción: insertar 11.



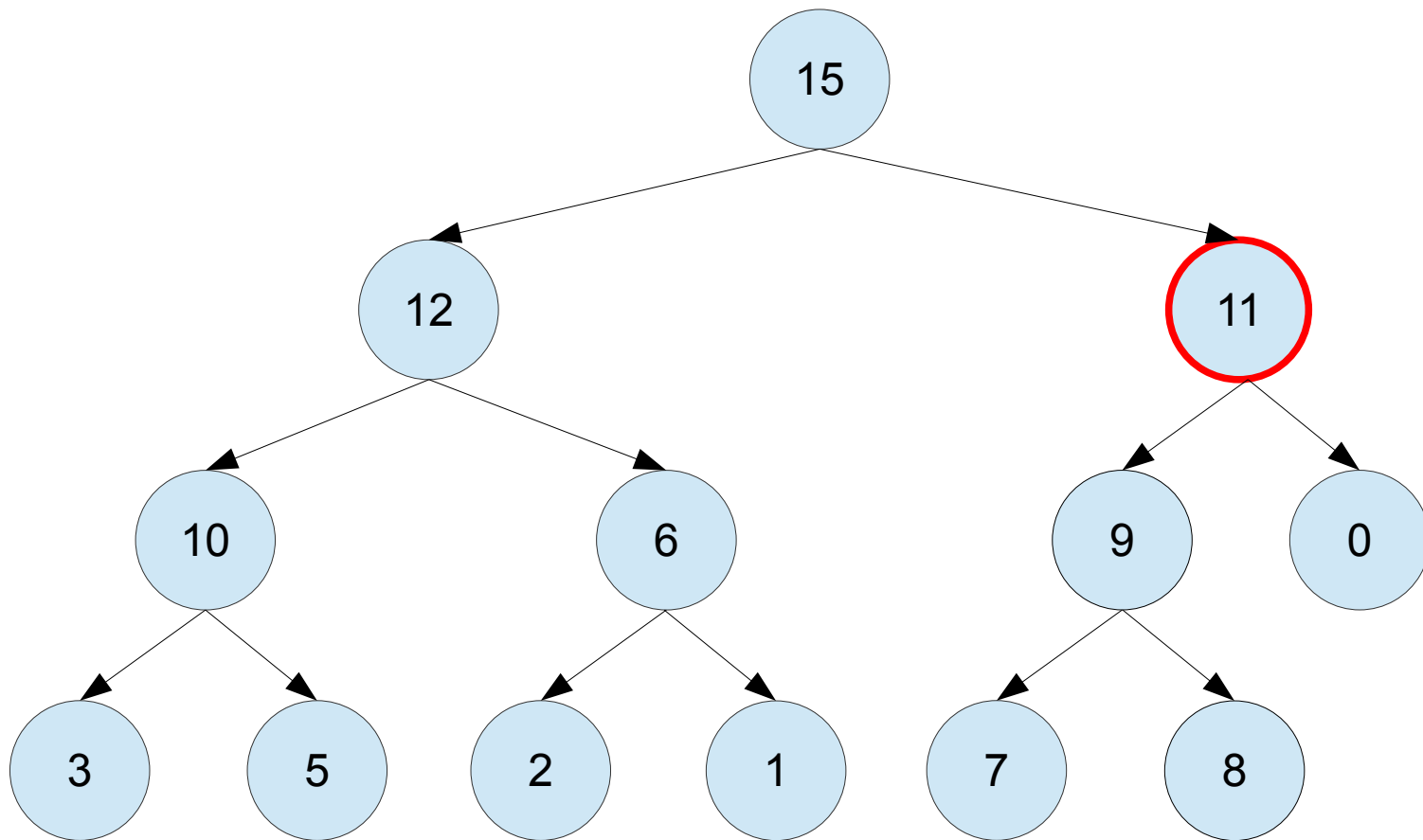
Montículo (*heap*)

- Inserción: insertar 11.



Montículo (*heap*)

- Inserción: insertar 11.



Montículo (*heap*)

- Eliminación:

Corresponde a eliminar el nodo de la raíz, y restaurar la propiedad de ordenamiento.

1. Reemplazar la raíz con el “último” elemento.

2. Comparar la nueva raíz con sus hijos:

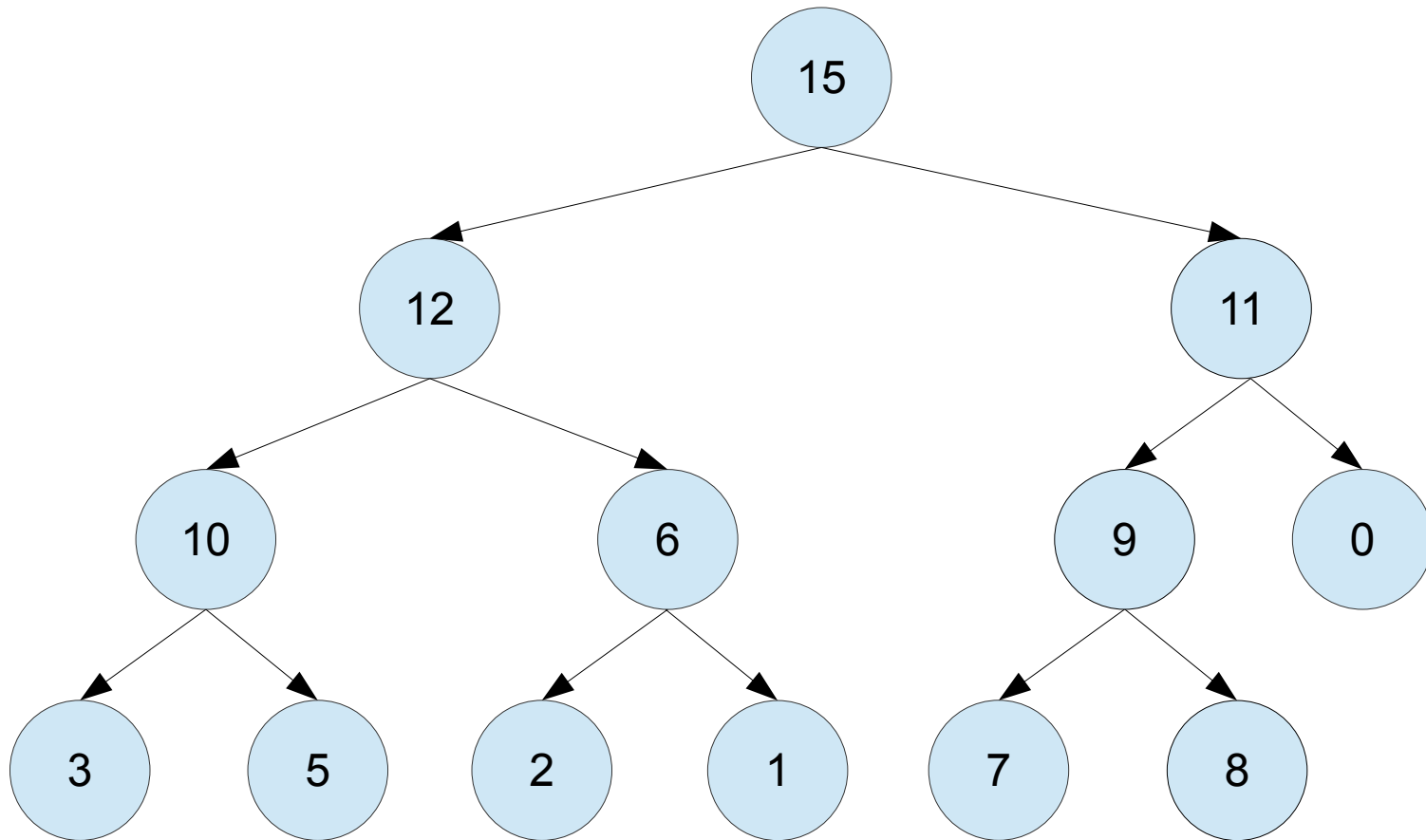
- 2.1 si están en el orden correcto, terminar.

- 2.2 si no, intercambiar el elemento con uno de sus hijos (el mínimo en *min heap* y el máximo en *max heap*) y retornar al paso 2.

Depende de la altura del montículo $\rightarrow O(\log n)$.

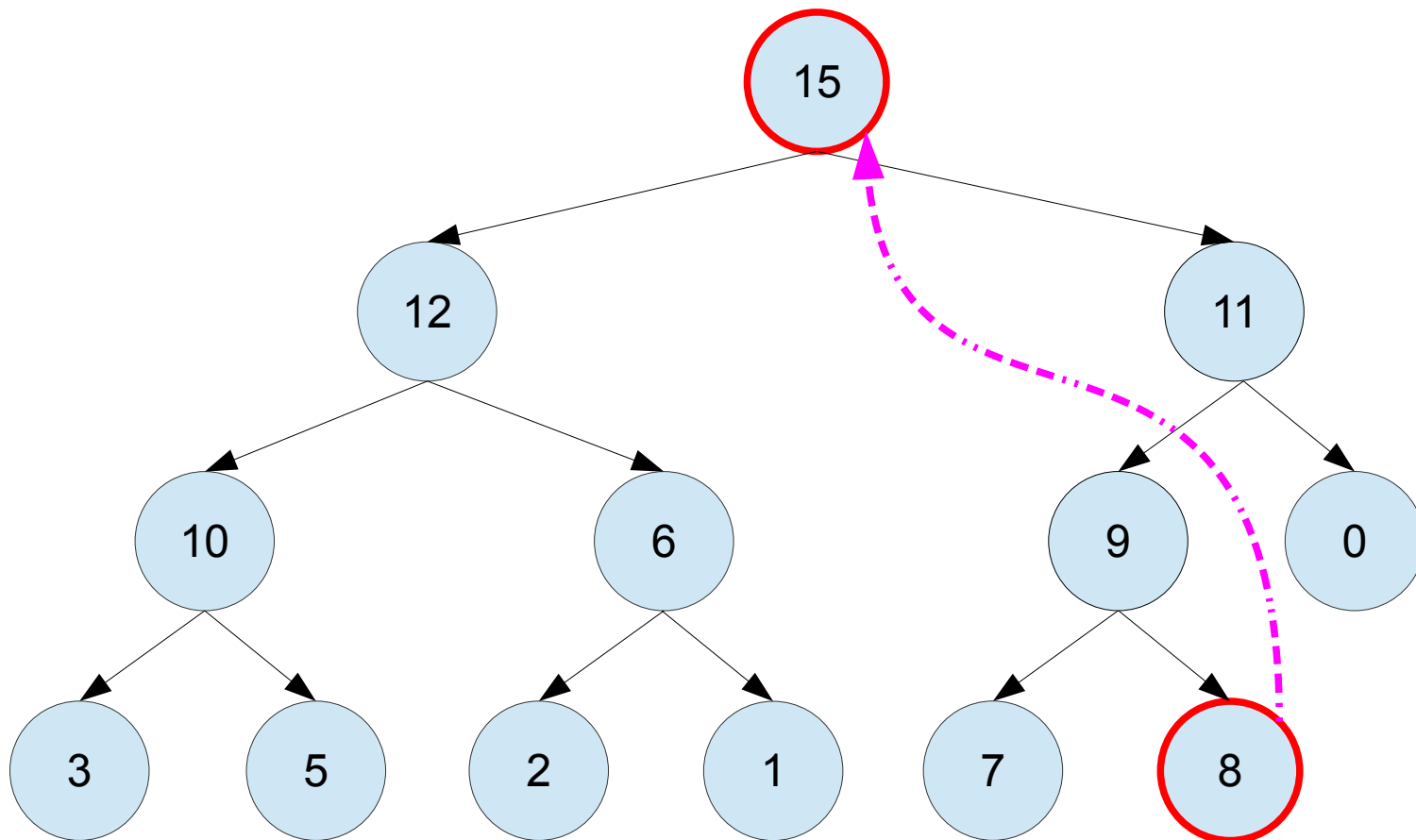
Montículo (*heap*)

- Eliminación:



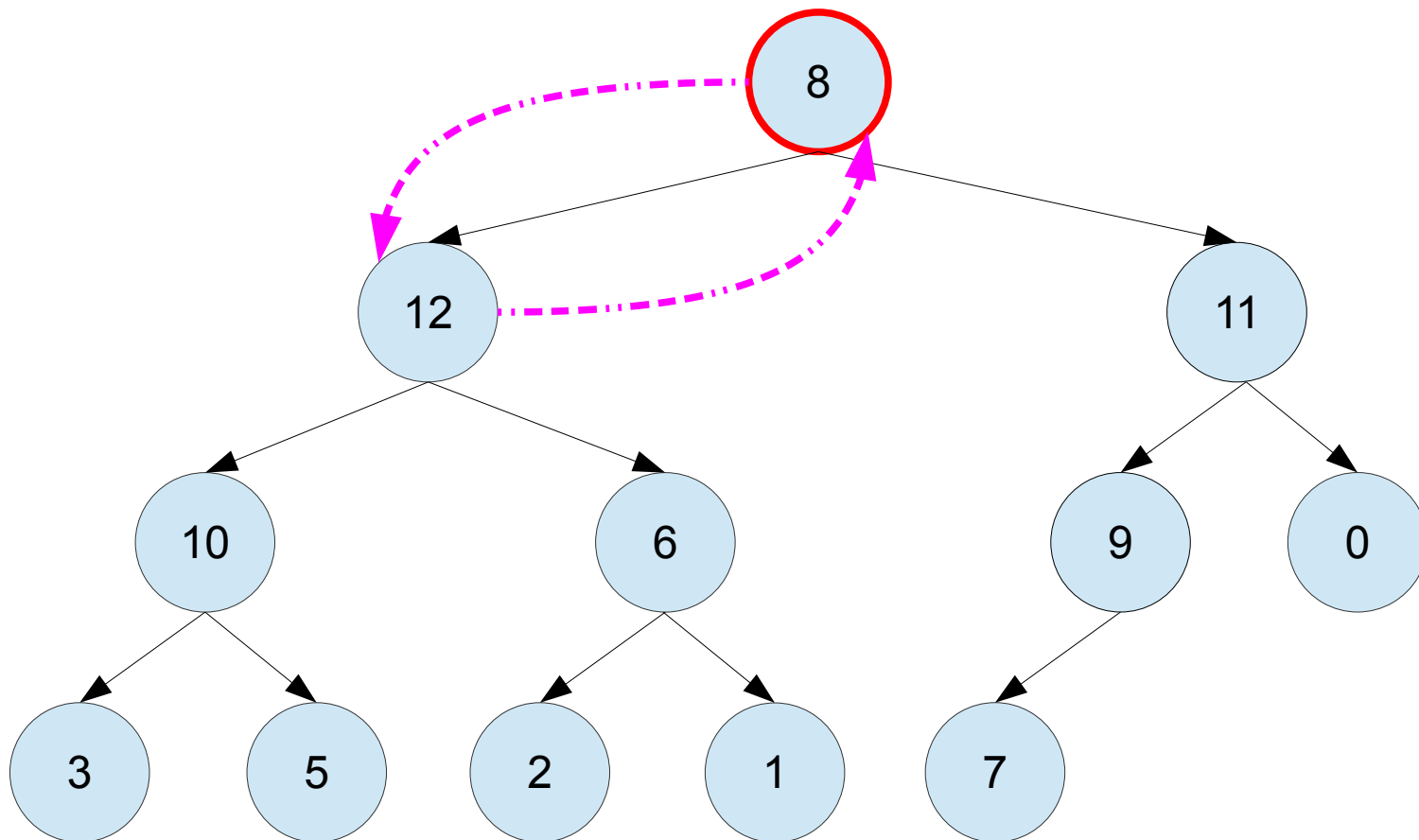
Montículo (*heap*)

- **Eliminación:**



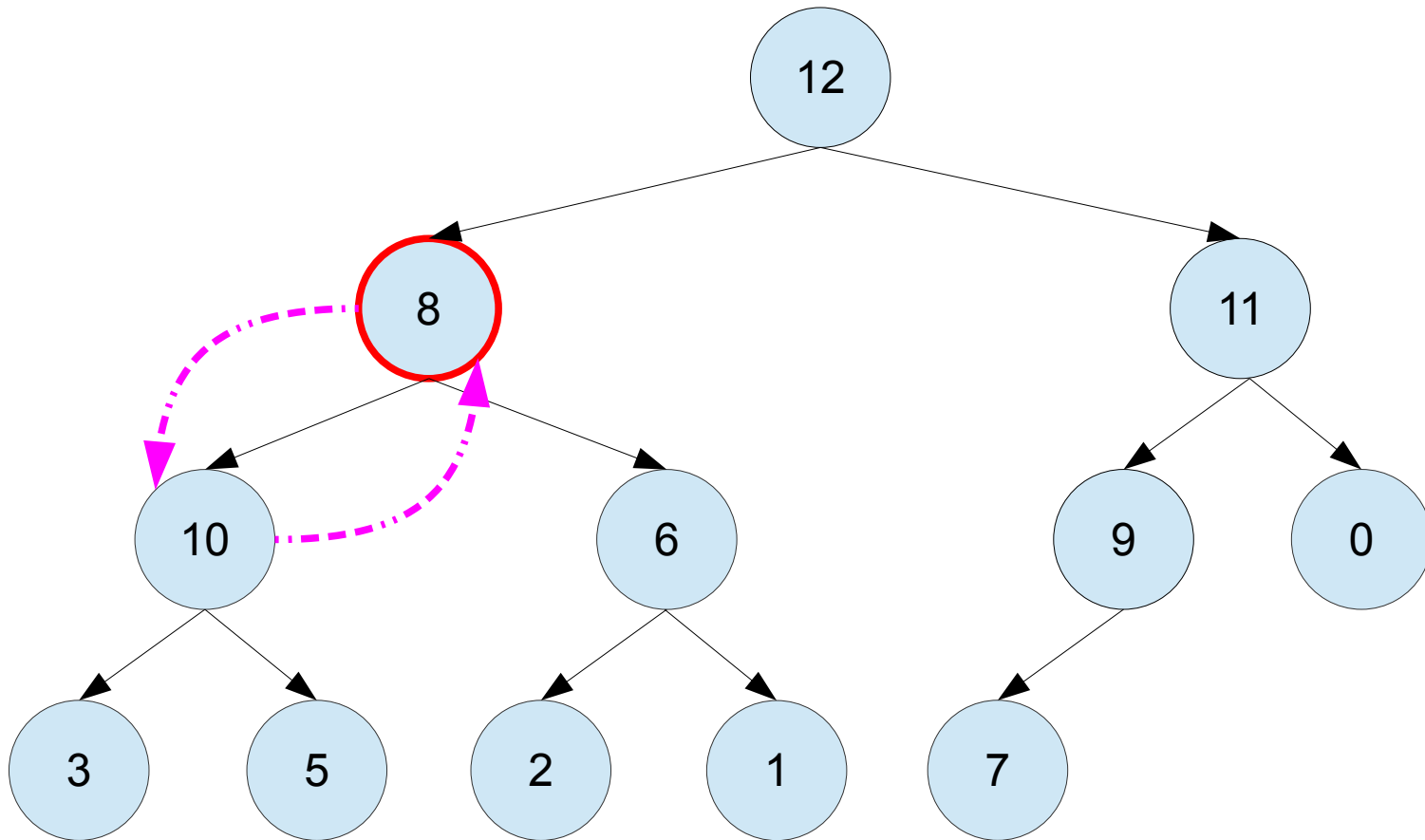
Montículo (*heap*)

- Eliminación:



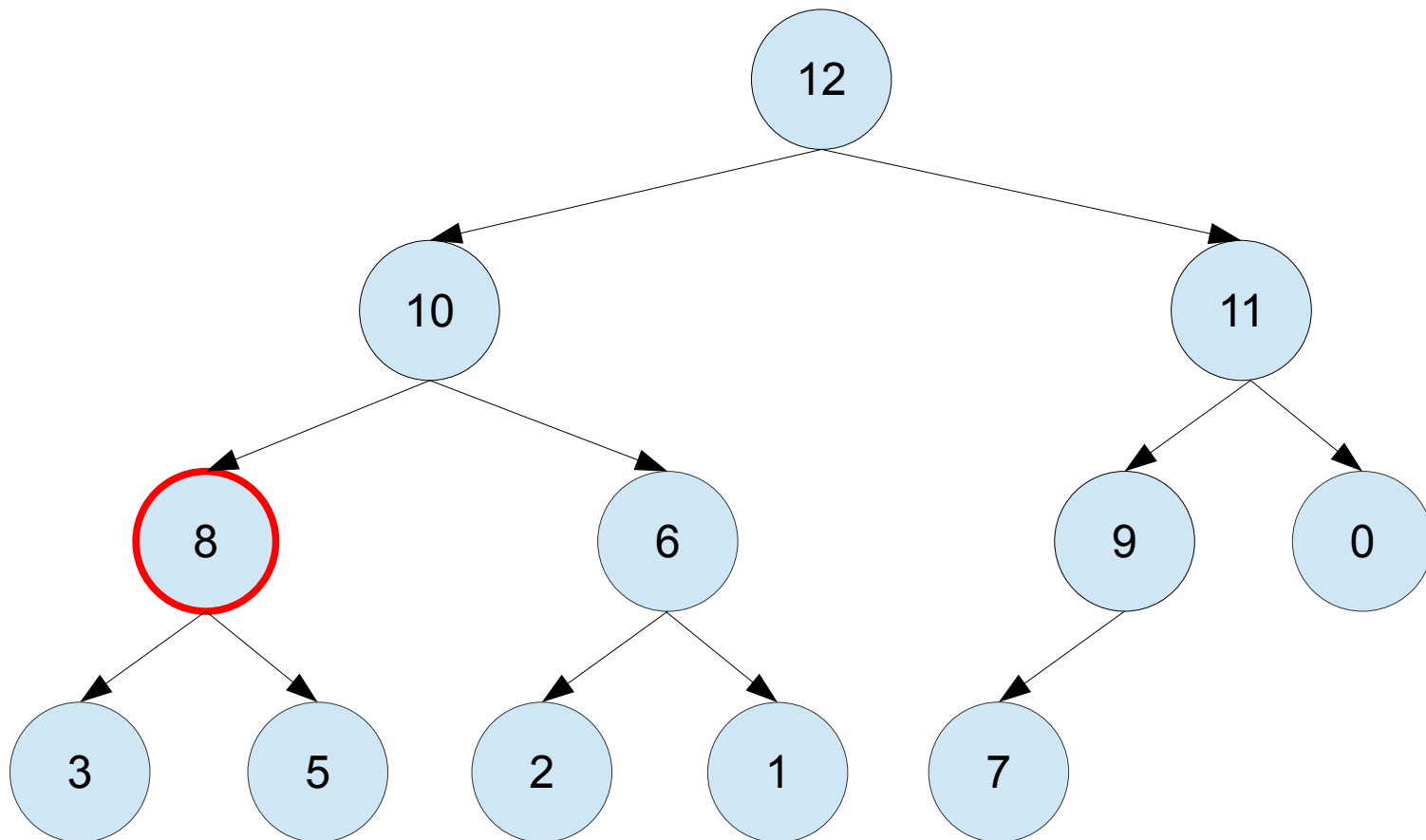
Montículo (*heap*)

- Eliminación:



Montículo (*heap*)

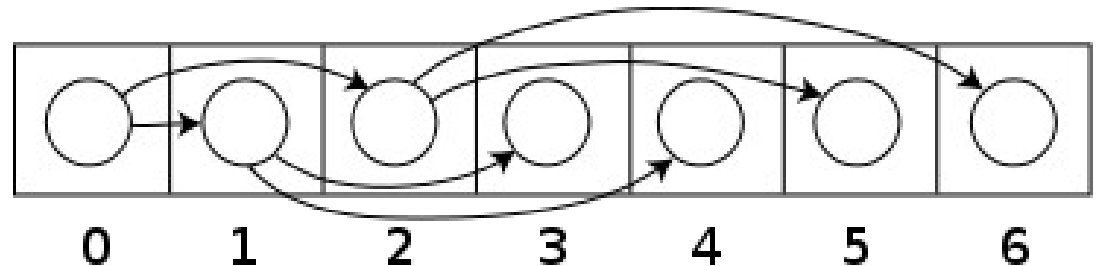
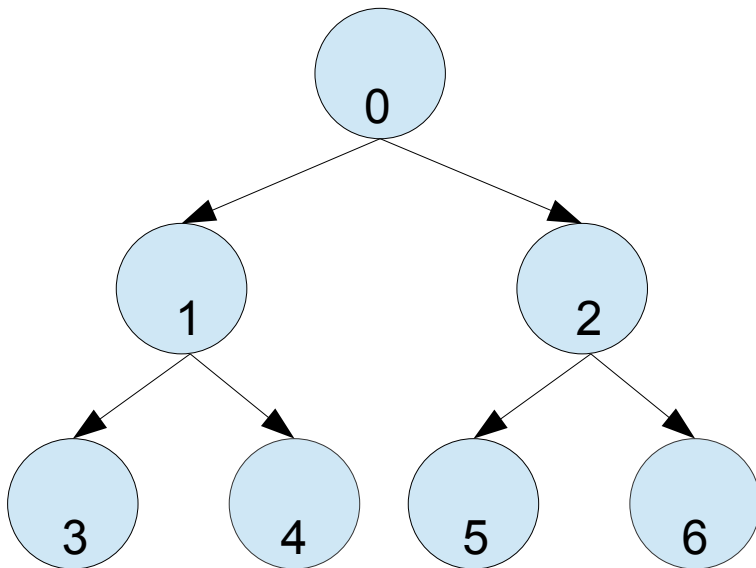
- Eliminación:



Montículo (*heap*)

- Implementación:

Representación utilizando un arreglo:



No se requieren apuntadores, sólo aritmética de índices dentro del arreglo.

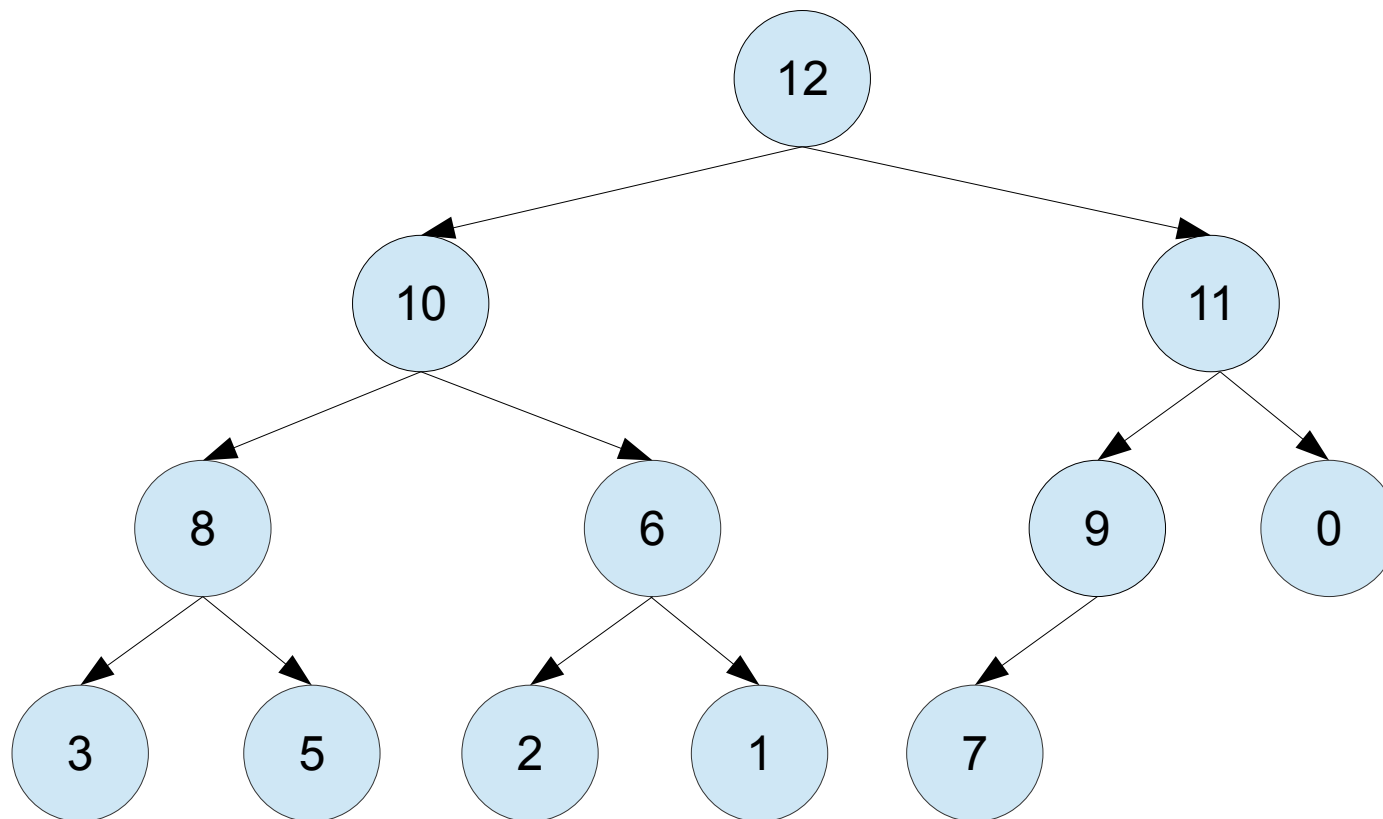
Montículo (*heap*)

- Implementación: aritmética de índices:
n: número de elementos en el montículo.
i: índice dentro del arreglo (0 hasta n-1).
Raíz: posición (índice) 0 del arreglo.
Cualquier elemento *a* con índice *i* tiene.
 - sus hijos en las posiciones $2i+1$ y $2i+2$.
 - su padre en la posición $\lfloor (i-1)/2 \rfloor$
($\lfloor \dots \rfloor$: función piso o parte entera).

Montículo (*heap*)

- Implementación:

12	10	11	8	6	9	0	3	5	2	1	7
----	----	----	---	---	---	---	---	---	---	---	---



Montículo (*heap*)

- Implementación:
usando cualquier contenedor de la STL.

```
#include <algorithm>

std::push_heap( it_begin, it_end )
std::pop_heap( it_begin, it_end )
std::is_heap( it_begin, it_end )
std::sort_heap( it_begin, it_end )
std::make_heap( it_begin, it_end )
```

Montículo (*heap*)

- Implementación: usando la STL.

Inserción: en el contenedor, luego en el montículo.

```
std::deque< T > vec;  
while( there_is_data )  
{  
    T new_data = get_next_data( );  
    vec.push_back( new_data );  
    std::push_heap( vec.begin( ), vec.end( ) );  
}
```

Montículo (*heap*)

- Implementación: usando STL:

Eliminación: del montículo, luego del contenedor.

```
std::deque< T > vec;  
std::pop_heap( vec.begin( ), vec.end( ) );  
T elim_data = vec.back( );  
vec.pop_back( );
```


Montículo (*heap*)

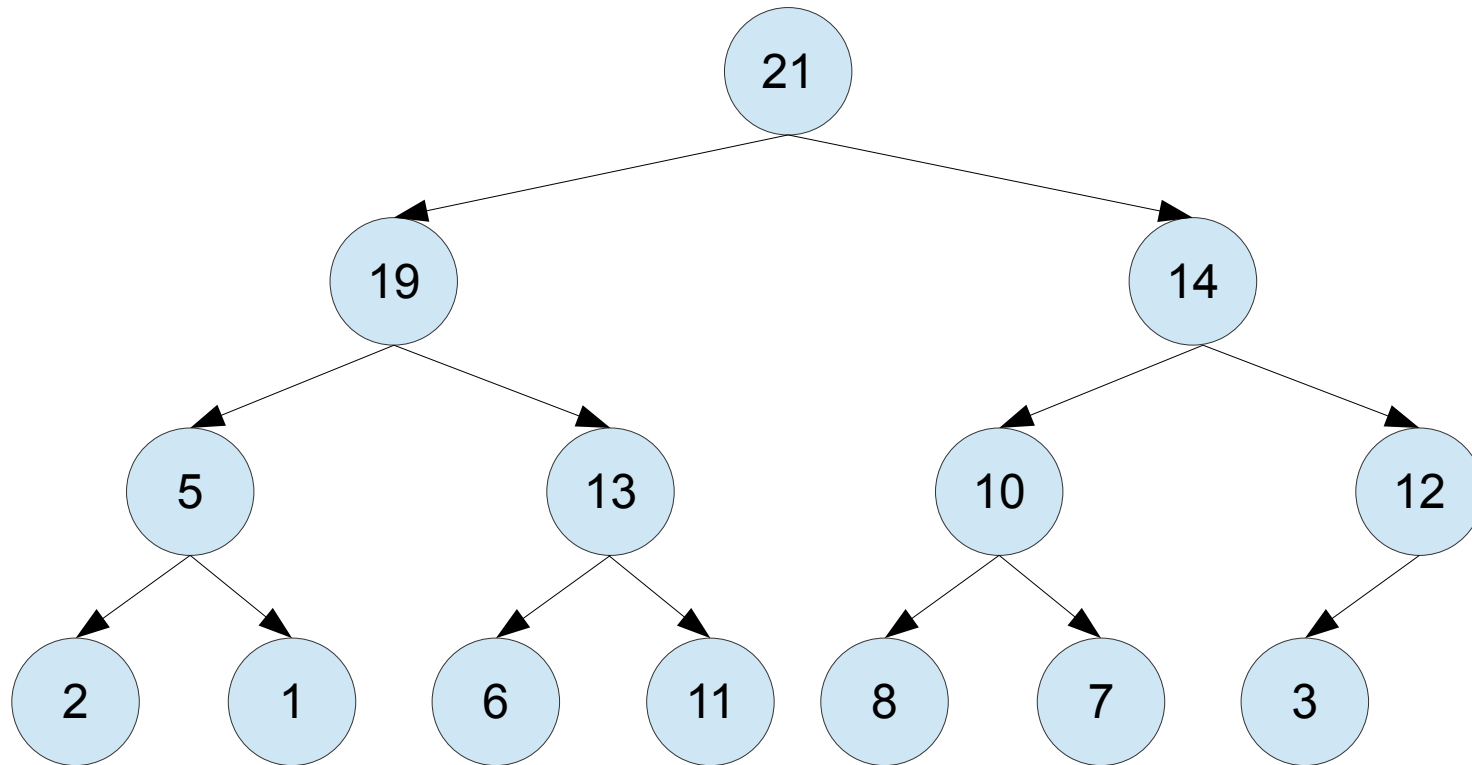
- Ejercicio:

Insertar los siguientes elementos en un montículo máximo:

12, 6, 14, 2, 13, 8, 19, 5, 1, 11, 21, 10, 7, 3

Montículo (*heap*)

21	19	14	5	13	10	12	2	1	6	11	8	7	3			
----	----	----	---	----	----	----	---	---	---	----	---	---	---	--	--	--



Montículo (*heap*)

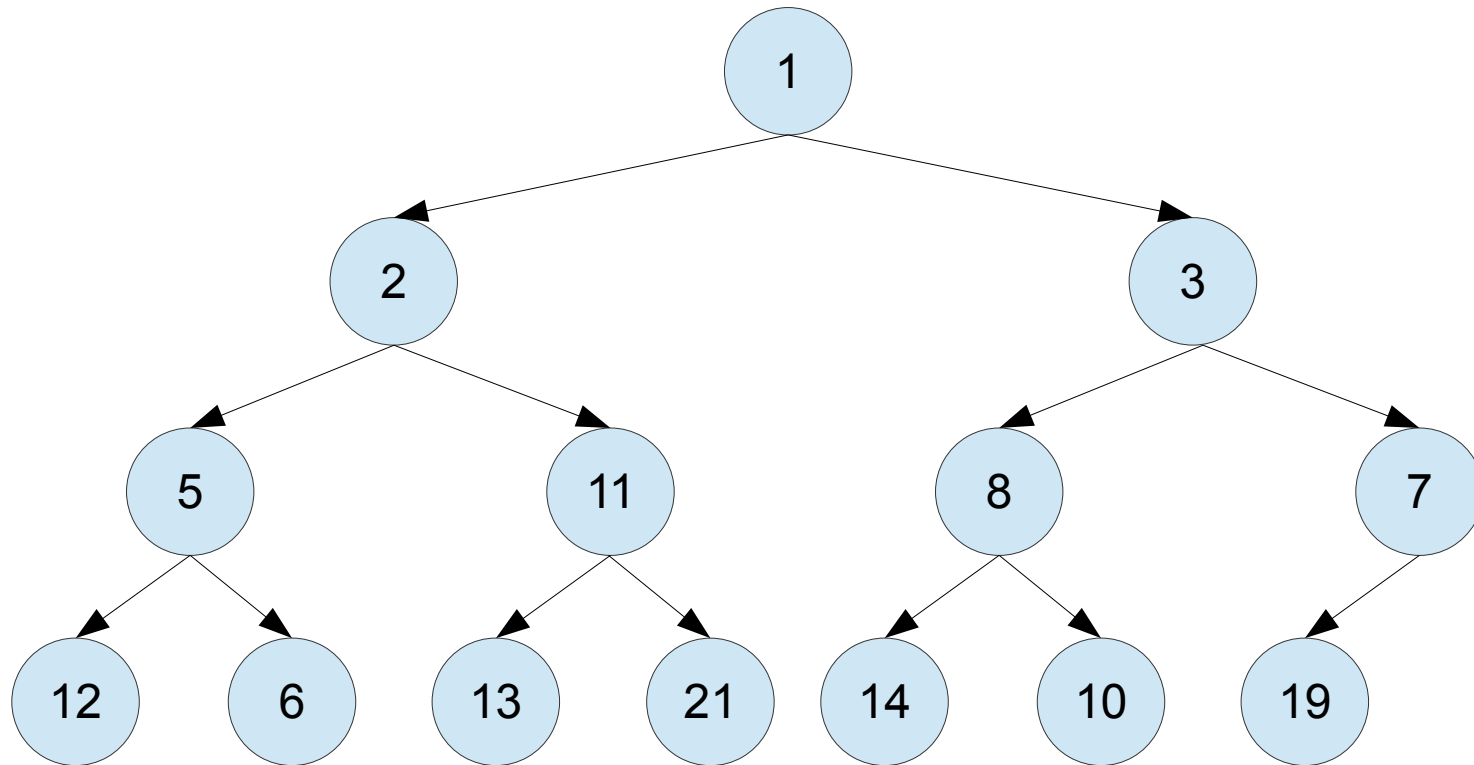
- Ejercicio:

Insertar los mismos elementos, ahora en un montículo mínimo:

12, 6, 14, 2, 13, 8, 19, 5, 1, 11, 21, 10, 7, 3

Montículo (*heap*)

1	2	3	5	11	8	7	12	6	13	21	14	10	19			
---	---	---	---	----	---	---	----	---	----	----	----	----	----	--	--	--



Árboles de Decisión

Árbol de Decisión

- Forma gráfica de representar y visualizar todos los eventos que pueden surgir a partir de la toma de decisiones secuenciales.
- Facilita la identificación de la decisión más acertada, a partir de probabilidades.
- Permite el análisis de consecuencias, del comportamiento en la toma de decisiones.

Árbol de Decisión




Ámbitos de utilización:

- Probabilidad y estadística:

Uso en investigación de operaciones, en teoría de la decisión.

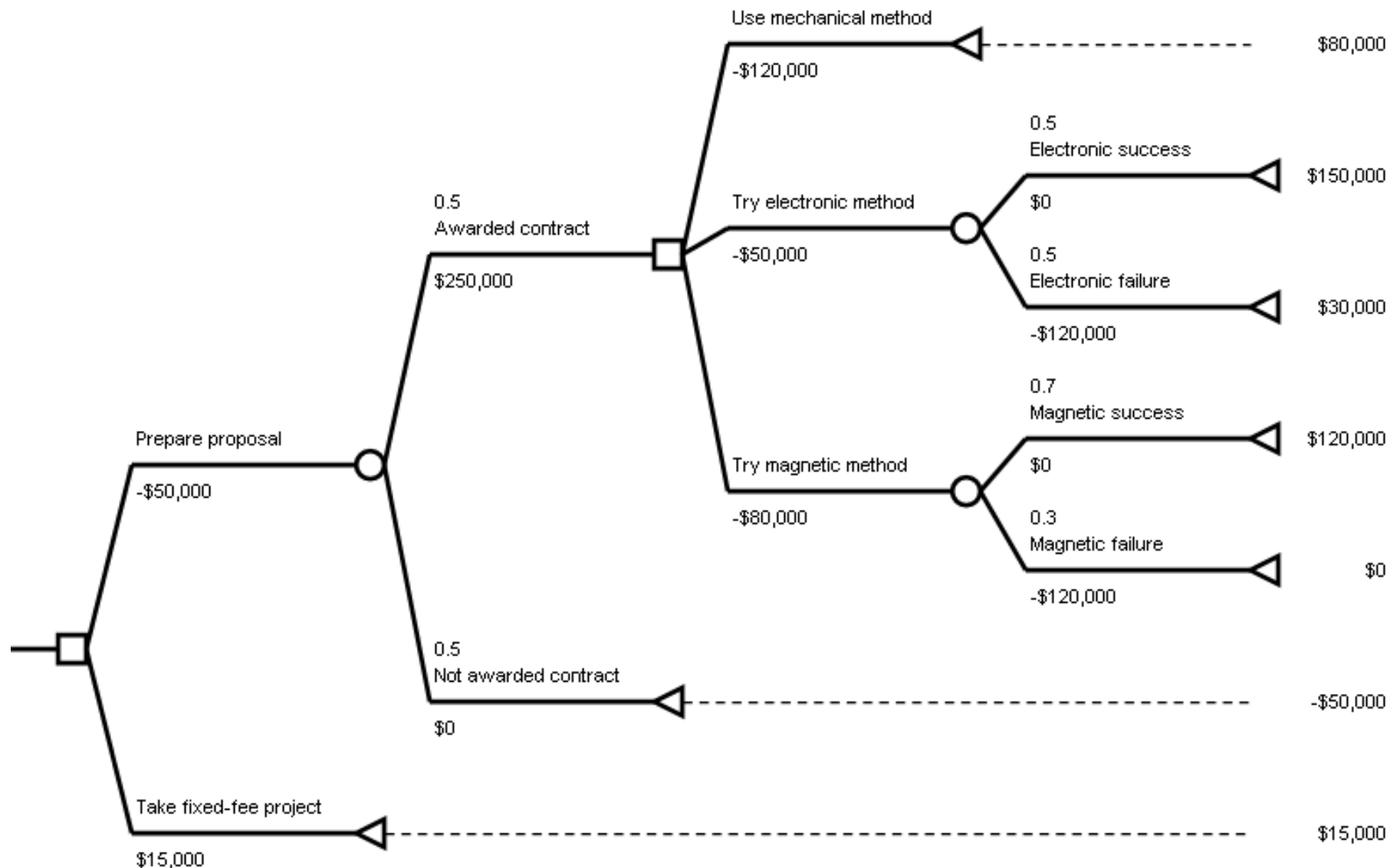
Herramienta de apoyo empresarial.

Tipos de nodos:

-  Nodo de decisión: indica una toma de decisión.
-  Nodo de probabilidad: indica un evento aleatorio.
-  Nodo terminal: posible salida o resultado.

Árbol de Decisión

TreePlan Decision Tree



Árbol de Decisión

Ámbitos de utilización: probabilidad y estadística.

- Análisis por medio de un árbol de decisión:

1. Definir el problema.
2. Dibujar el árbol de decisión.
3. Asignar probabilidades a los eventos aleatorios.
4. Estimar los resultados para cada combinación posible de alternativas.
5. Resolver el problema utilizando la ruta que proporcione la política óptima.

Árbol de Decisión

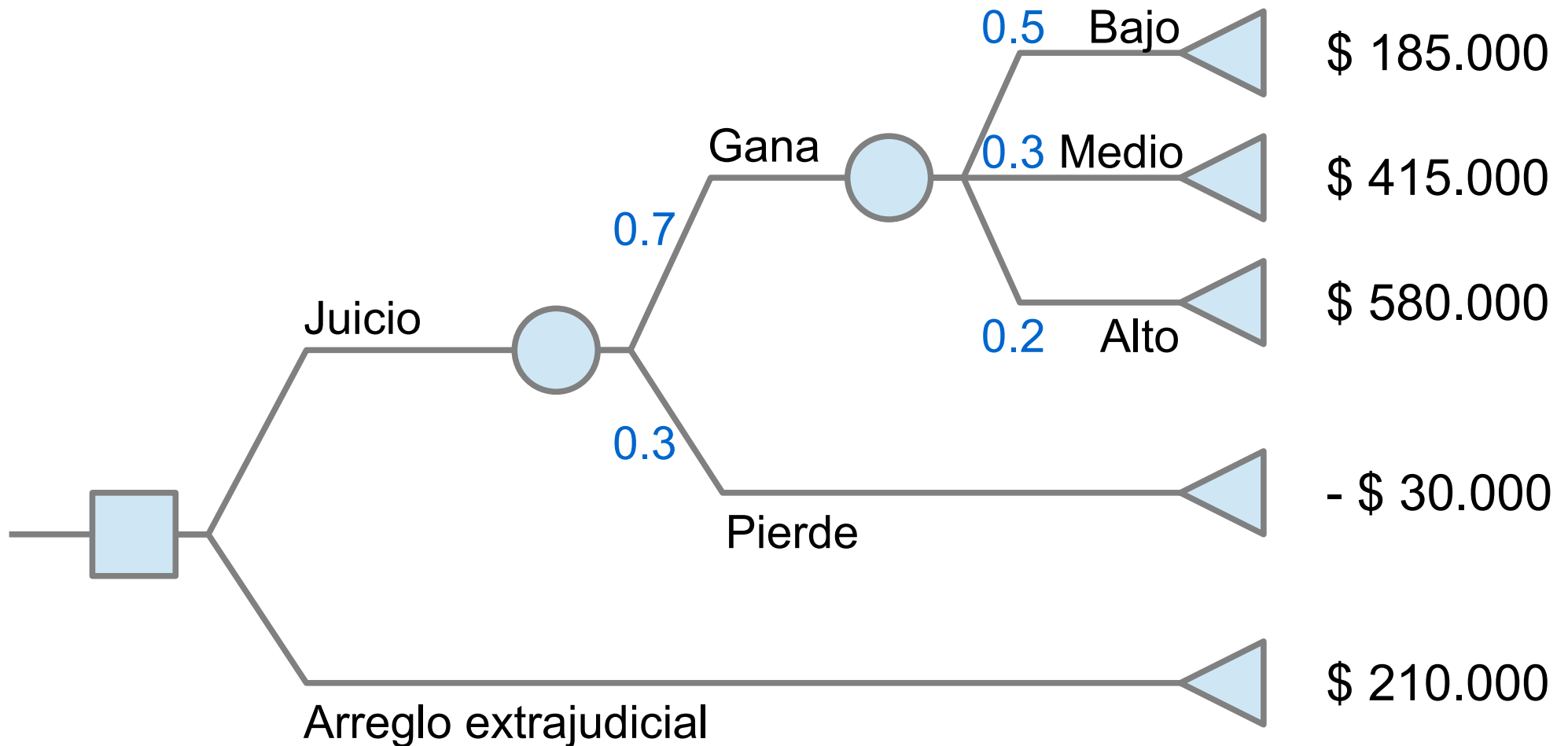
- Ejemplo:

Una compañía de seguros nos ofrece una indemnización por accidente de \$210.000. Si no aceptamos la oferta y decidimos ir a juicio podemos obtener \$185.000, \$415.000 o \$580.000 dependiendo de las alegaciones que el juez considere aceptables. Si perdemos el juicio, debemos pagar los costos que ascienden a \$30.000.

Sabiendo que el 70% de los juicios se gana, y de éstos, en el 50% se obtiene la menor indemnización, en el 30% la intermedia y en el 20% la más alta, determinar la decisión más acertada.

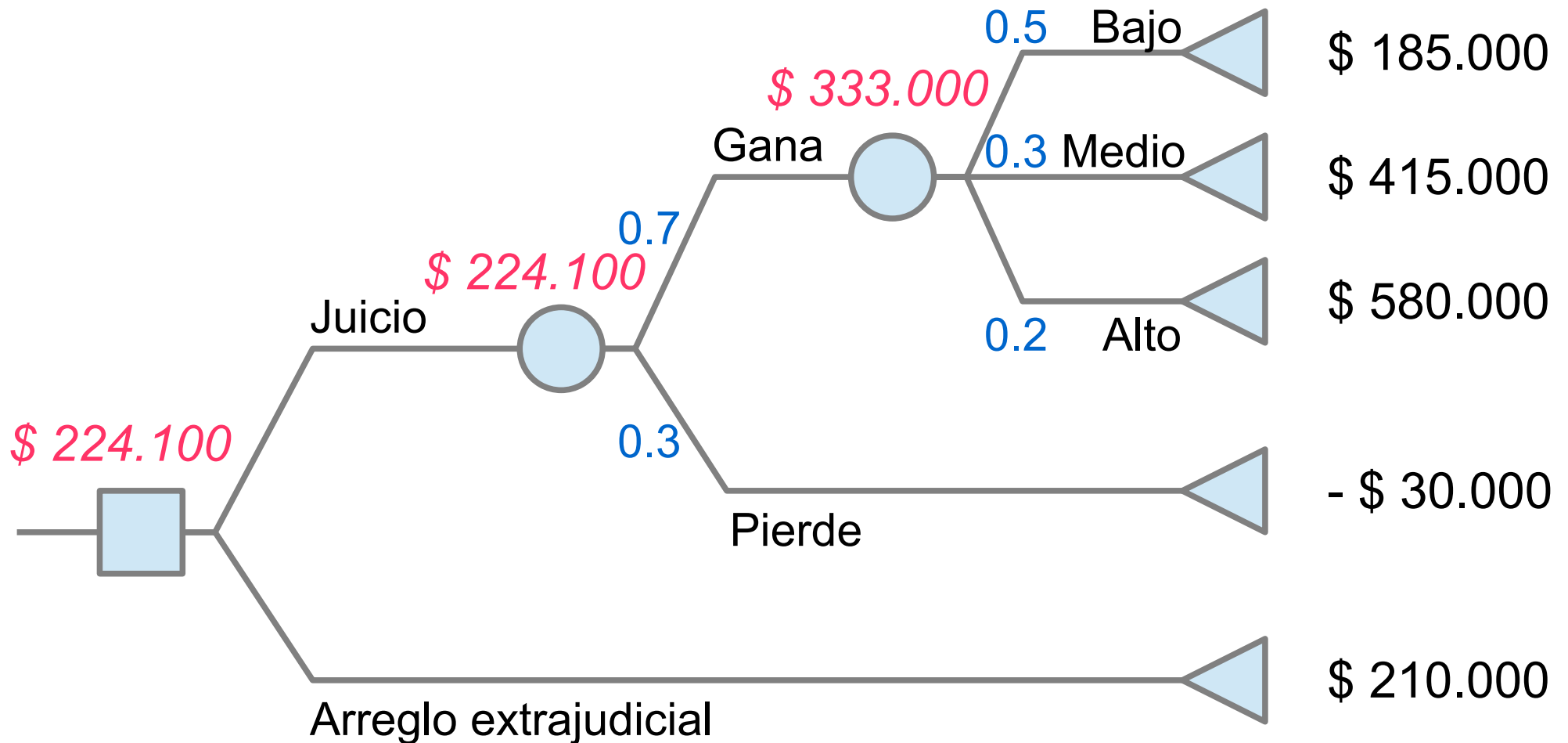
Árbol de Decisión

- Ejemplo:



Árbol de Decisión

- Ejemplo:



Árbol de Decisión

Ámbitos de utilización:

- Aprendizaje de máquina y sistemas expertos:

Enseñar al computador a tomar decisiones de acuerdo a comportamientos previos.

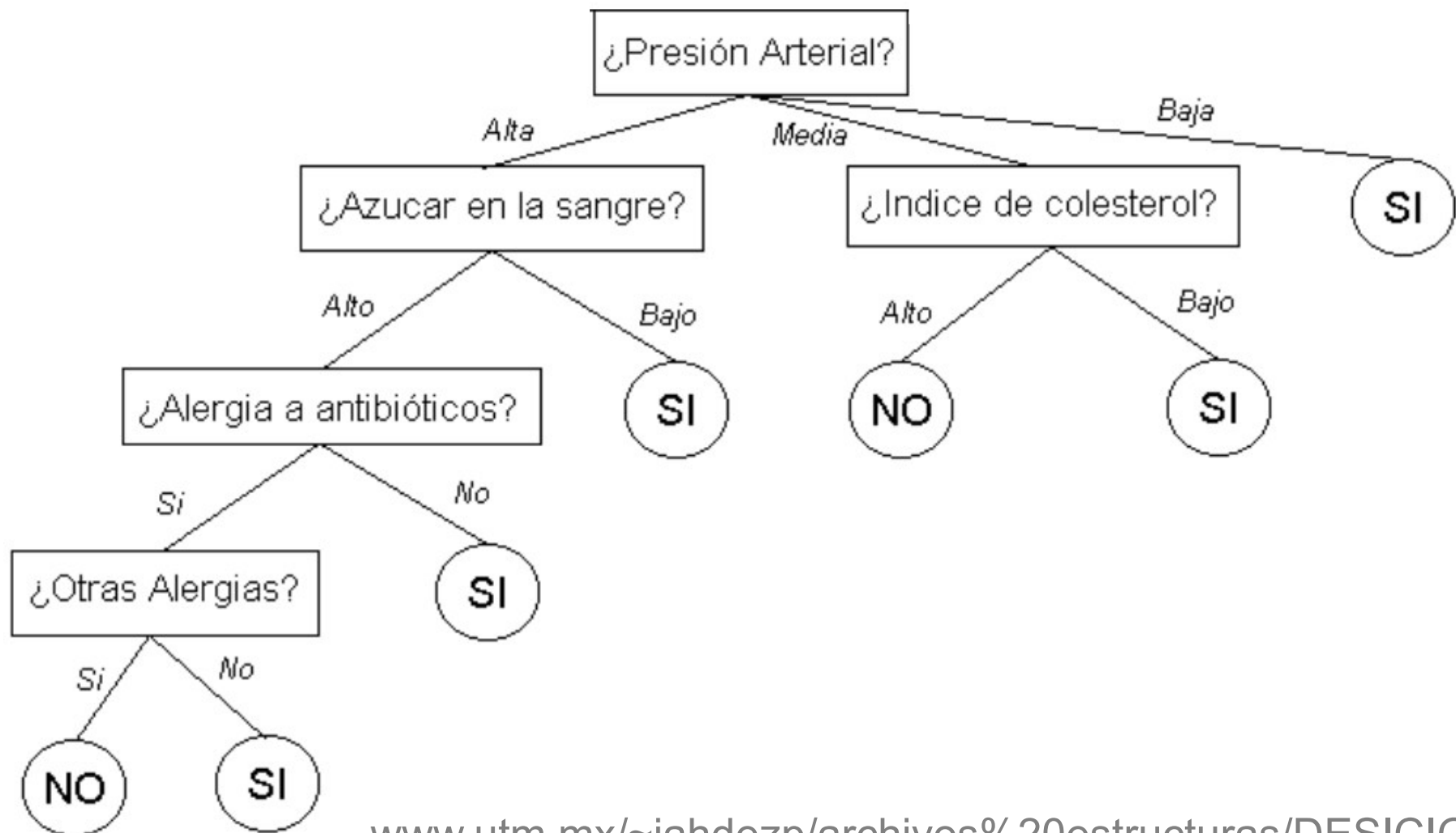
Realizar predicciones (clasificación, regresión).

Tipos de nodos:

- Nodos internos: implementan una función de decisión.
- Nodos hoja: salidas esperadas o posibles.

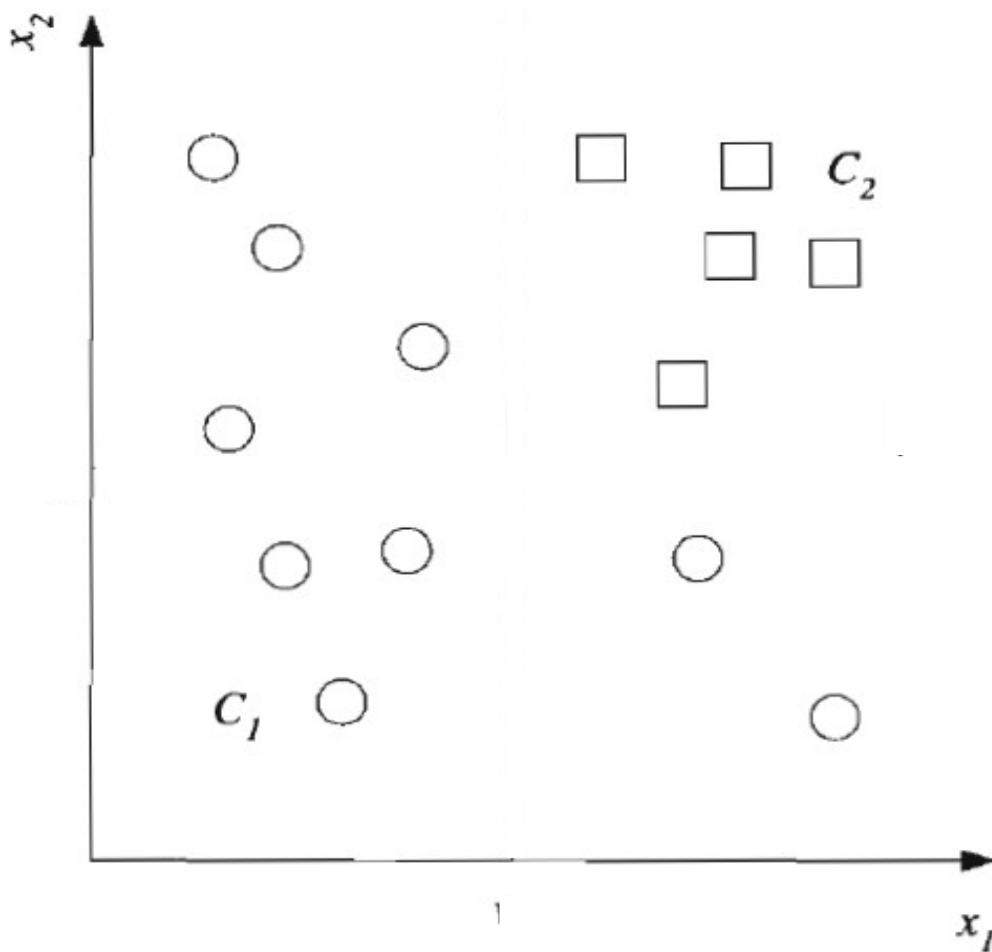
Árbol de Decisión

- Ejemplo: Administración de un fármaco.



Árbol de Decisión

- Ejemplo: Árbol de decisión a partir de datos.

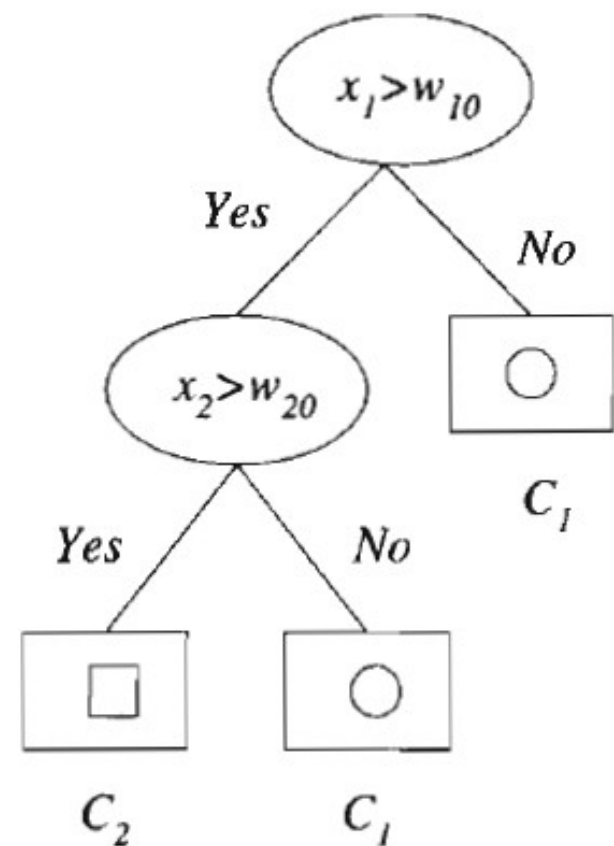
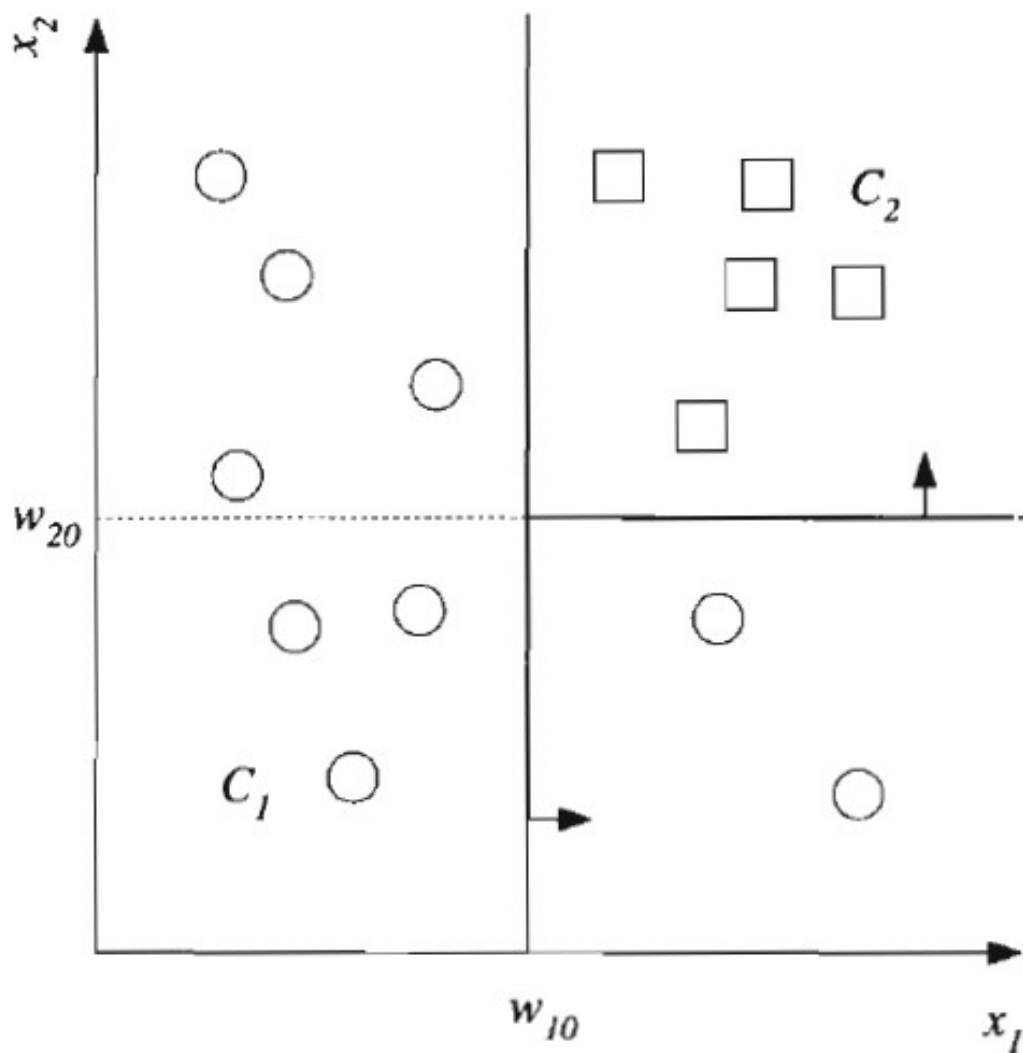


Cada dato
caracterizado por:
- Dos valores
numéricos (x_1 y x_2)
- Una forma
geométrica

¿Cómo separar las
diferentes formas?

Árbol de Decisión

- Ejemplo: Árbol de decisión a partir de datos.



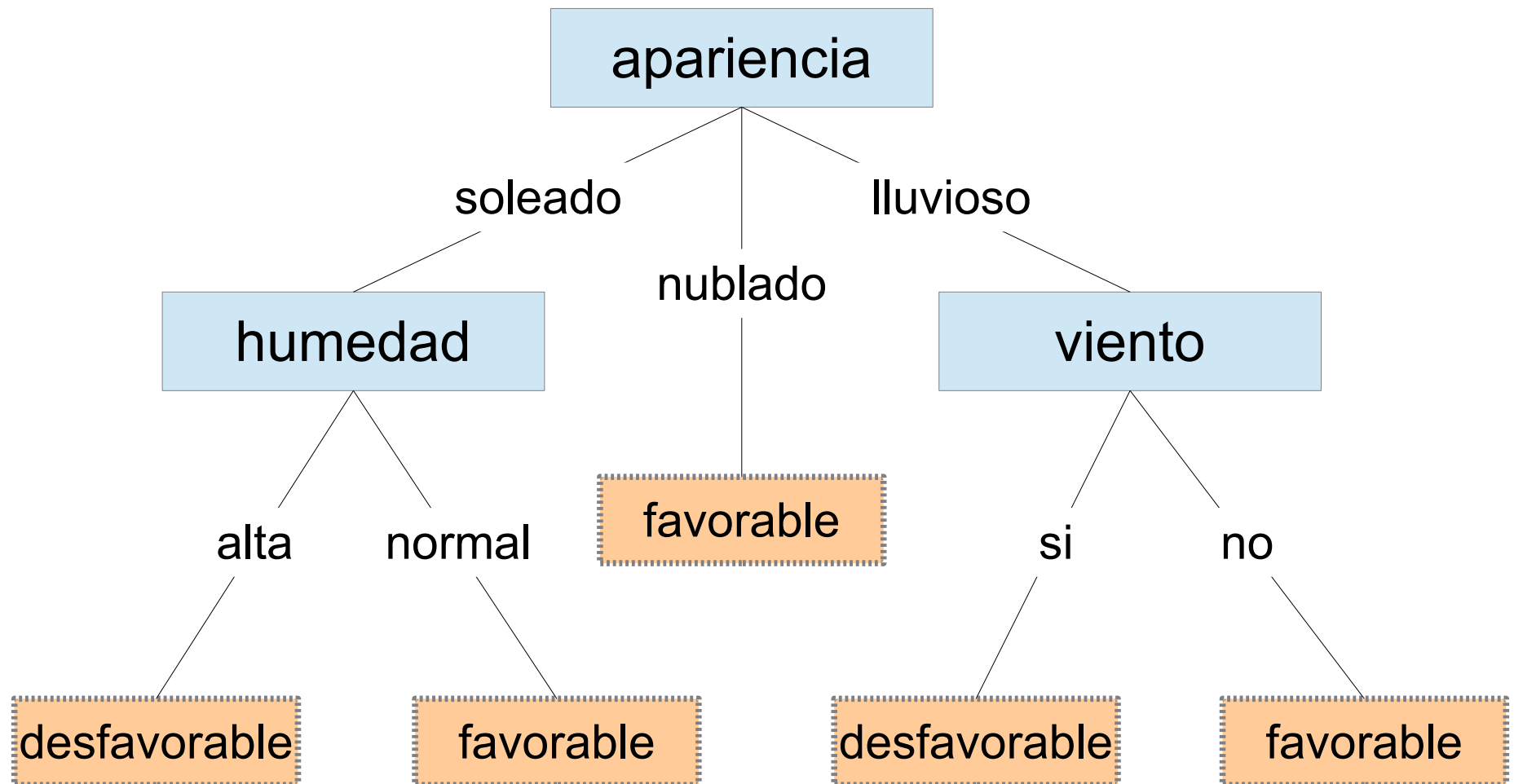
Árbol de Decisión

- Ejemplo de clasificación: 1. entrenamiento.

No.	Atributos				Clase
	Apariencia	Temperatura	Humedad	Viento	
1	soleado	alta	alta	no	desfavorable
2	soleado	alta	alta	si	desfavorable
3	nublado	alta	alta	no	favorable
4	lluvioso	media	alta	no	favorable
5	lluvioso	baja	normal	no	favorable
6	lluvioso	baja	normal	si	desfavorable
7	nublado	baja	normal	si	favorable
8	soleado	media	alta	no	desfavorable
9	soleado	baja	normal	no	favorable
10	lluvioso	media	normal	no	favorable
11	soleado	media	normal	si	favorable
12	nublado	media	alta	si	favorable
13	nublado	alta	normal	no	favorable
14	lluvioso	media	alta	si	desfavorable

Árbol de Decisión

- Ejemplo de clasificación: 2. árbol de decisión.



Referencias

- en.wikipedia.org/wiki/Decision_tree
- www.dmae.upct.es/~mcruiz/Telem06/Teoria/arbol_decision.pdf
- en.wikipedia.org/wiki/Decision_tree_learning
- www.utm.mx/~jahdezp/archivos%20estructuras/DESICION.pdf
- www.dmi.unict.it/~apulvirenti/agd/Qui86.pdf
- Alpaydin, Ethem. Introduction to machine learning. MIT press, 2004.

Referencias

- en.wikipedia.org/wiki/Decision_tree
- en.wikipedia.org/wiki/Decision_tree_learning