Algoritmos y Estructuras de Datos 2015

AVL

Ejercicio 1 - Anexo

Había una vez un chimpancé llamado *Luchu Bandor,* cuyo significado era “Mono Playboy”. *Luchu* estaba infelizmente casado con *Bunty Mona,* una chimpancé muy bonita pero de baja estatura. *Luchu* era alto y guapo, se sentía incómodo cuando estaba con *Bunty* en lugares públicos, ya que la gente los miraba a ellos continuamente. En un momento dado, *Luchu* no pudo soportar más esta situación y decidió hacer justicia a su nombre. Él comenzó a buscar una nueva esposa en el “Colegio Nacional de Señoritas Chimpancés”. Cada día *Luchu* se subía a unas cañas de bambú y esperaba a que el ejercicio matutino empezara. Desde allí podía ver a todas las chimpancés haciendo su rutina de ejercicio diario. Ahora, *Luchu* estaba buscando a **una chimpancé más alta pero que sea más baja que él**, y también estaba interesado en **aquella chimpancé un poco más alta que él**. Sin embargo, **alguien de su misma altura no la consideraba**.

Luchu pudo modelar la situación descripta a través de un **árbol AVL**, el cuál contenía todas las alturas de las señoritas chimpancés que él había observado durante un cierto período de tiempo.

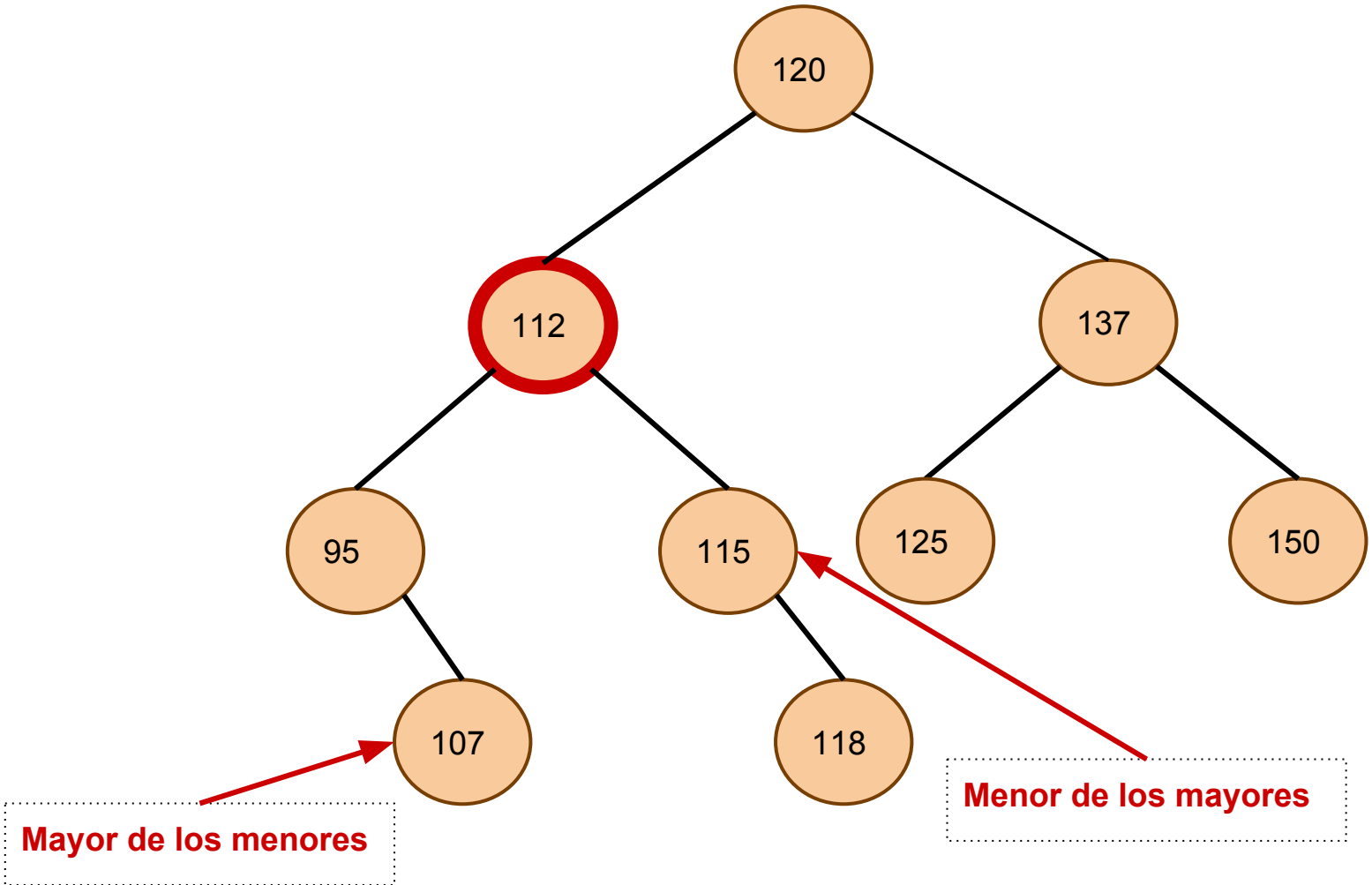
Su trabajo consiste en ayudar a *Luchu* para **encontrar a las dos mejores chimpancés de acuerdo al criterio de selección establecido: la chimpancé más alta de las más bajas que él y la más baja entre las más altas que él.**

Usted debe implementar un método en la clase árbol AVL, considerando que **recibe como parámetro la altura de *Luchu*** y debe **devolver las alturas de las dos chimpancés buscadas ordenados de manera creciente**. **En el caso que sea imposible encontrar alguna de estas dos alturas devuelva un valor igual a 0 para la menor y 999 para la mayor.** (0 y 999 no son alturas válidas, no están en el árbol)

Importante: considere que en el árbol existe una altura igual a la altura de *Luchu*.

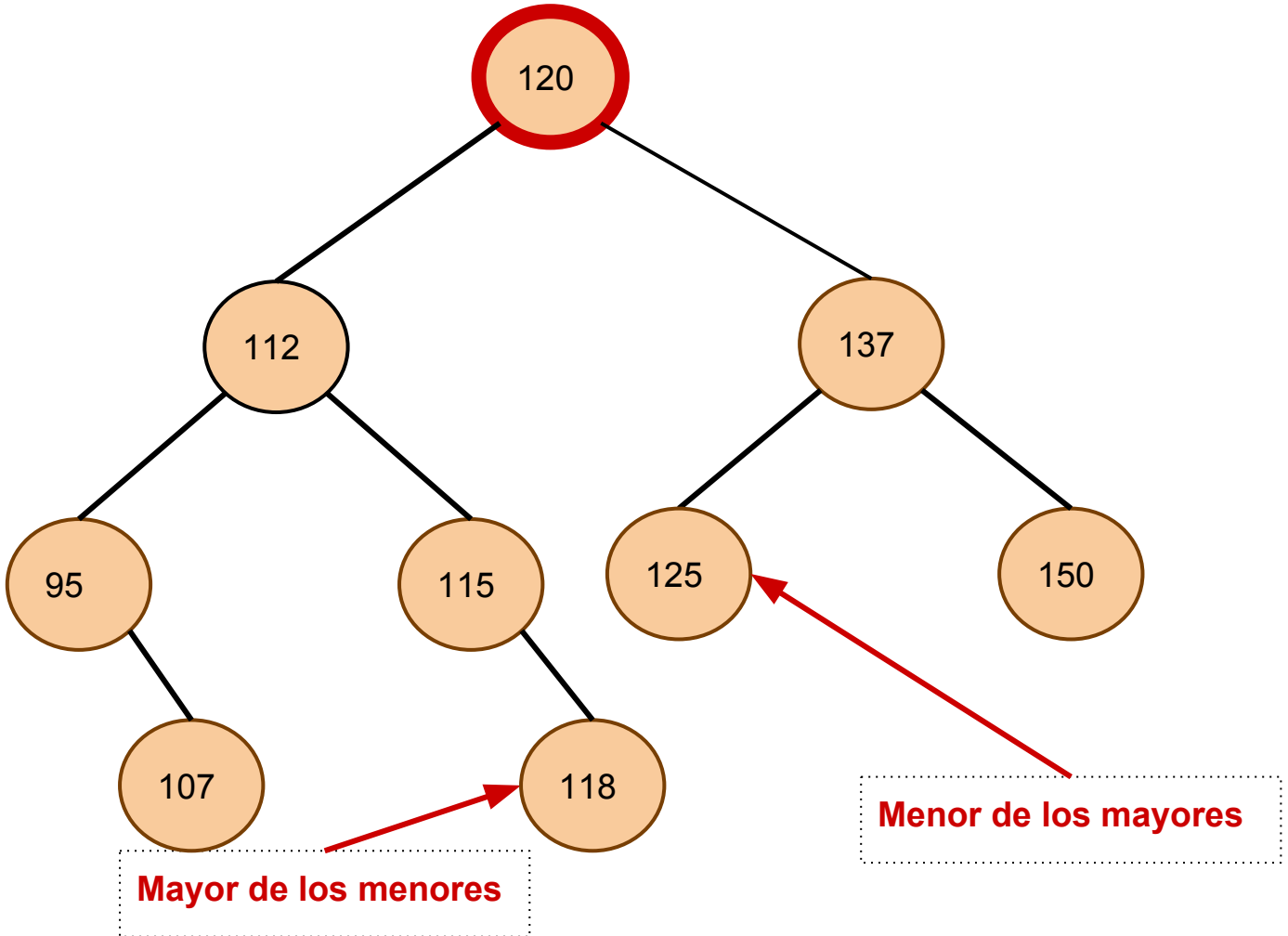
Casos a tener en cuenta

Altura de Luchu: 112



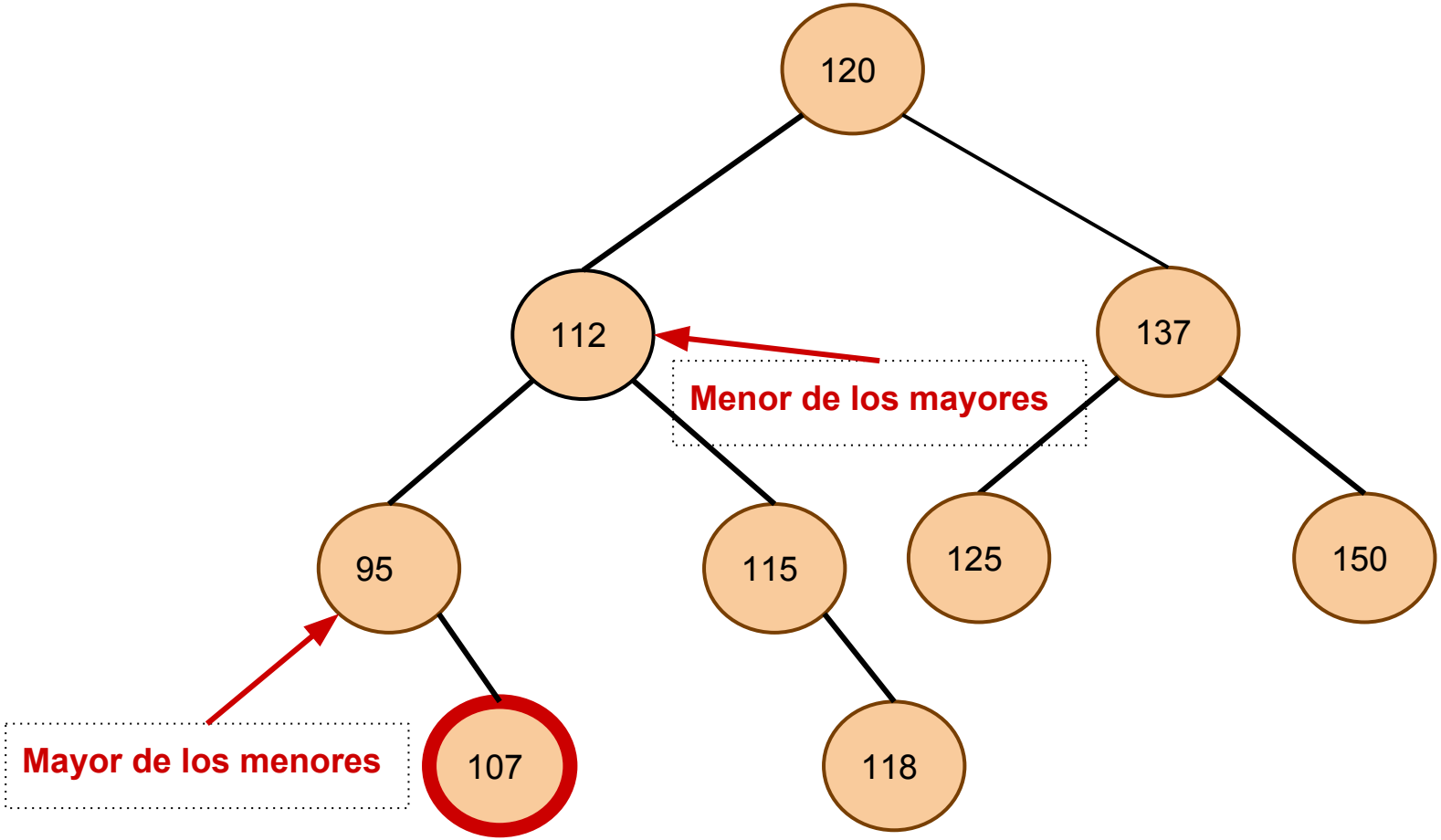
Casos a tener en cuenta

Altura de Luchu: 120



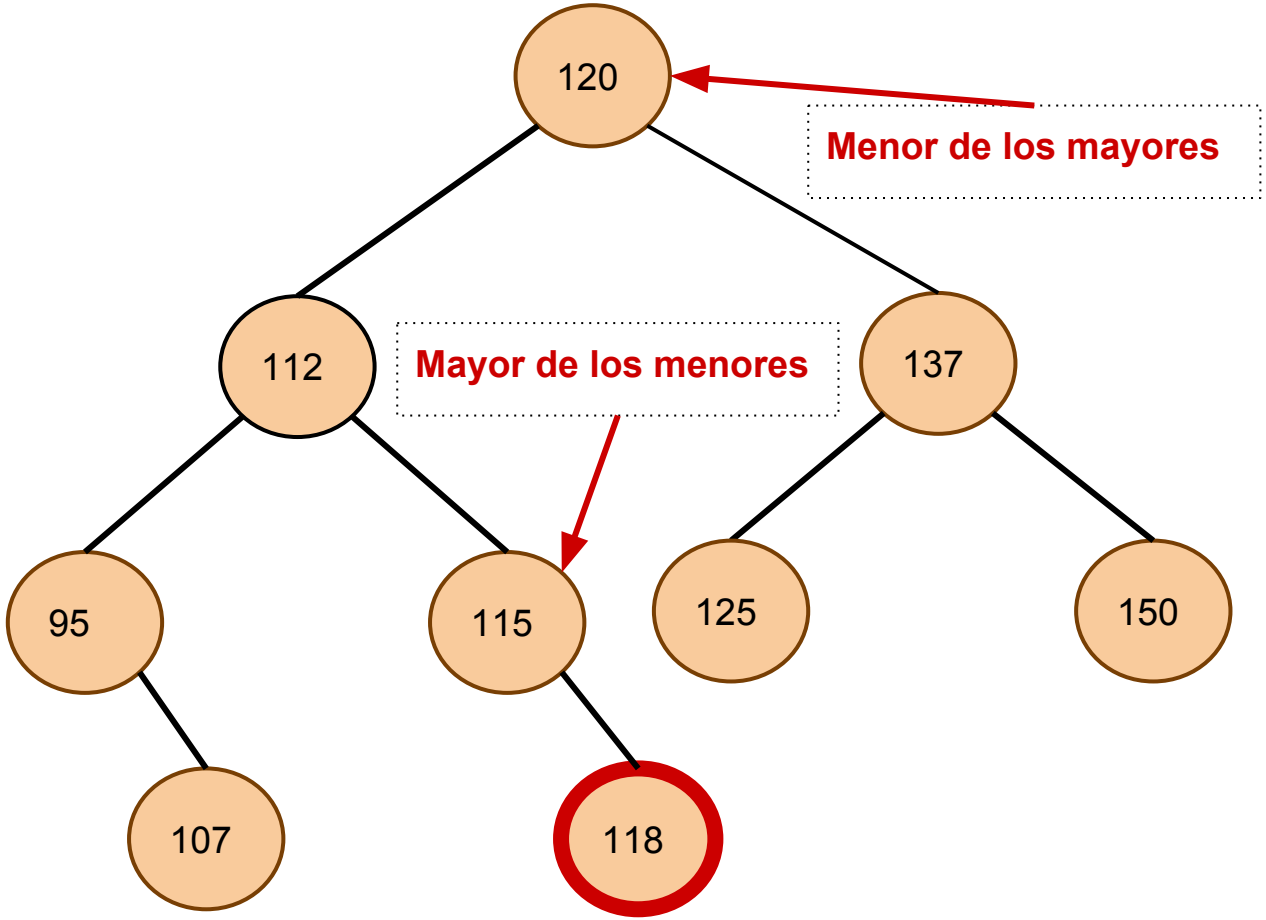
Casos a tener en cuenta

Altura de Luchu: 107



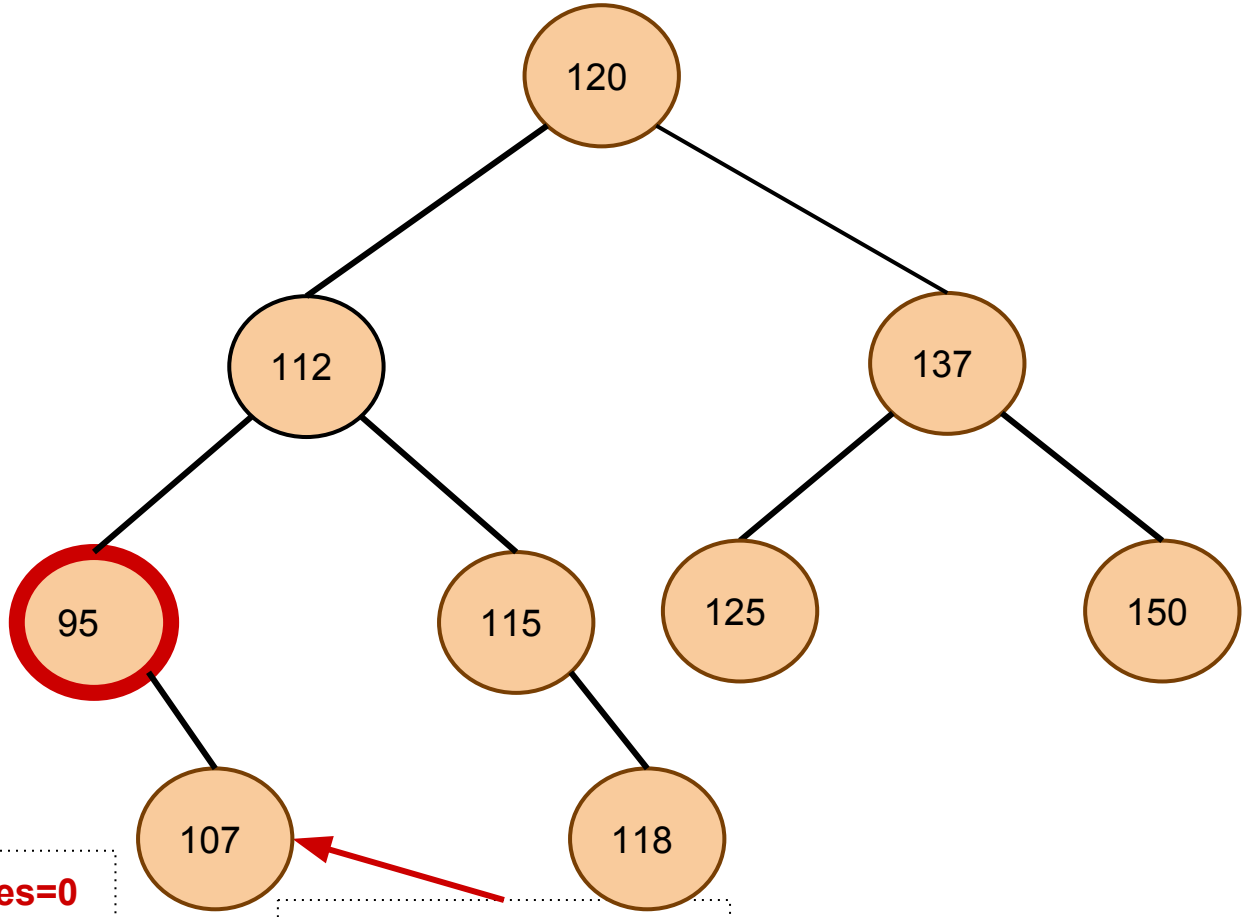
Casos a tener en cuenta

Altura de Luchu: 118



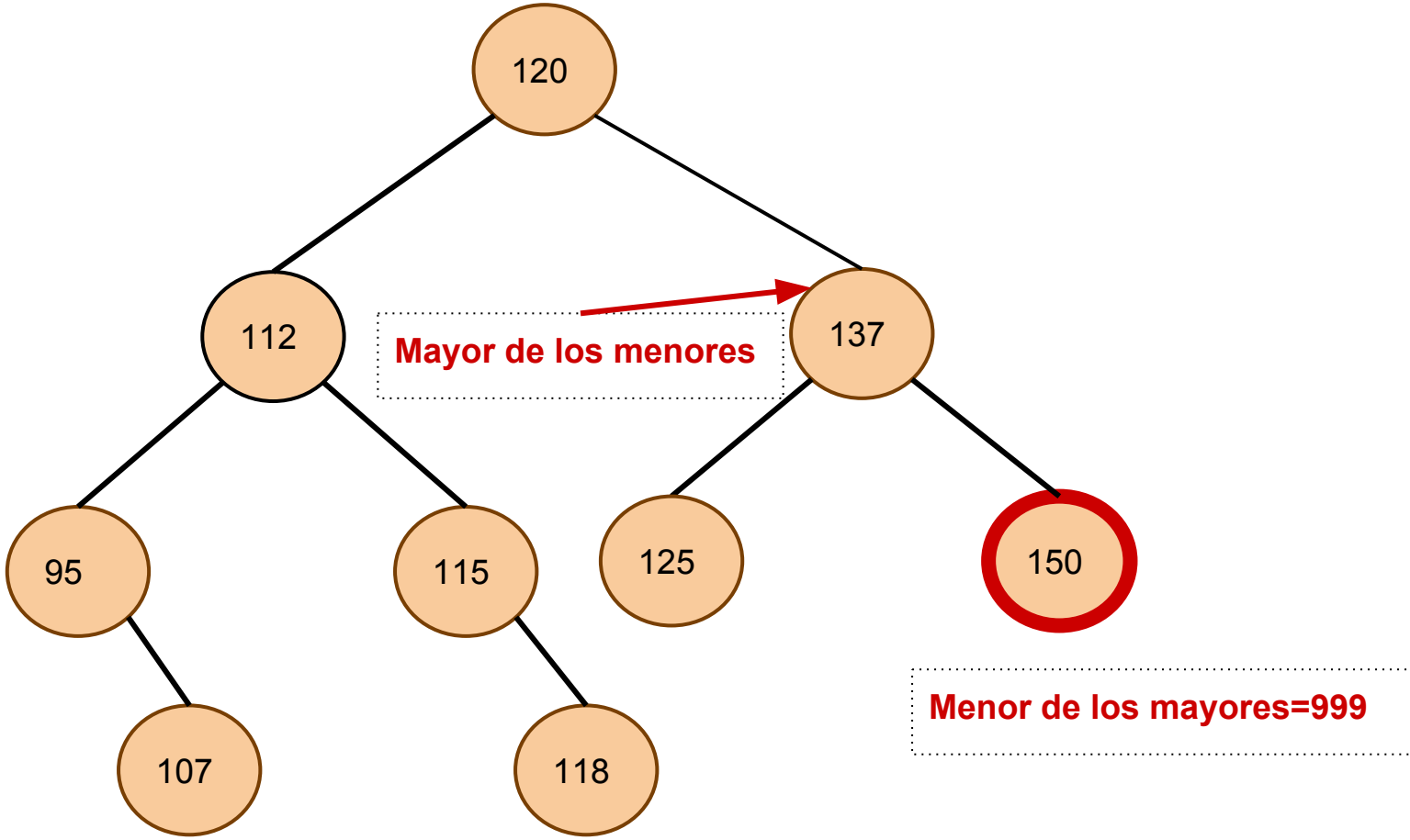
Casos a tener en cuenta

Altura de Luchu: 95



Casos a tener en cuenta

Altura de Luchu: 150



Solución

```
private void  find (Integer altLuchu) {
if (!this.esVacio()) {
    if (altLuchu.compareTo(this.getDatoRaiz()) == 0)  {                    // encontré la altura de Luchu !
              if( ! this.getHijoIzquierdo().esVacio())
                    altMenor  = ((BuscaAlturas)this.getHijoIzquierdo()).findMax();        // busco la altura mayor
                                                                                          //del sub. izq.
              if( ! this.getHijoDerecho().esVacio())
                    altMayor = ((BuscaAlturas)this.getHijoDerecho()).findMin();   // busco la altura menor del sub.
                                                                                          //der.
    }else {
              if (altLuchu.compareTo(this.getDatoRaiz())  > 0) {
                    ((BuscaAlturas)this.getHijoDerecho()).find(altLuchu);       // busco en la rama derecha
                    if (altMenor == 0)                                          // verifica si se enc. la mas chica
                        altMenor  = this.getDatoRaiz();                         // si no encontró, la mas chica inmediata es la 1er
                                                                                // encontrada al volver del llamado recursivo
              } else {
                   ((BuscaAlturas)this.getHijoIzquierdo()).find(altLuchu);      // busco en la rama izquierda
                   if (altMayor == 999)                                         // verifica si se enc. la mas gde
                        altMayor  = this.getDatoRaiz();                         //sino encontro, la mas gde. inmediata es la 1er
                                                                                // encontrada al volver del llamado recursivo
              }
    }
}
```

Solución(cont.)

```
//busca la altura mayor del sub-árbol izquierdo
private Integer findMax() {
   Integer alt;
   if( ! this.getHijoDerecho().esVacio())
         alt = ((BuscaAlturas)this.getHijoDerecho()).findMax();
   else {
         alt = this.getDatoRaiz();
   }
    return alt;
}
```

Solución(cont.)

```
// busca la altura menor del sub-árbol derecho
private Integer findMin() {
     Integer alt;
    if( ! this.getHijoIzquierdo().esVacio())
                alt = ((BuscaAlturas)this.getHijozquierdo( )).
    findMin();
    else {
          alt = this.getDatoRaiz();
    }
    return alt;
}
```

Solución(cont.)

```
public class BuscaAlturas extends ArbolAVL<Integer> {

    public static Integer altMayor  = 0;
    public static Integer altMenor = 999;

    public ListaGenerica<Integer> buscar (Integer alt){
        ListaGenerica<Integer> l = new ListaEnlazadaGenerica<Integer>();
        this.find(alt );
        l.agregar( altMenor, 1);          // agrega a la lista l los valores
                                          // hallados, ordenados de manera
                                          // creciente
        l.agregar( altMayor, 2);
        return l;
    }

    .....
}
```