3.2. Arbres AVL

DEFINICIONS (I)

- L'eficiència en la cerca d'un element en un arbre binari de cerca es mesura en termes de:
  - Nombre de comparacions
  - Alçària de l'arbre
- Arbre completament equilibrat: els elements de l'arbre han d'estar repartits en el mateix nombre entre el subarbre esquerre i el dret, de tal forma que la diferència en nombre de nodes entre els dos subarbres siga com a màxim 1
- Problema: el manteniment de l'arbre
- Arbres AVL: desenvolupats per Adelson-Velskii i Landis (1962). Els AVL són arbres equilibrats respecte a l'altura dels subarbres:

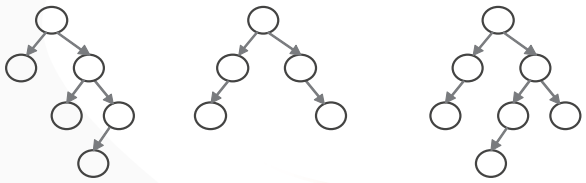
  "Un arbre està equilibrat respecte a l'altura si i només si per a cada un dels nodes ocorre que les altures dels dos subarbres difereixen com a màxim en 1"
- Conseqüència 1. Un arbre buit està equilibrat respecte a l'altura
- Conseqüència 2. L'arbre equilibrat òptim serà aquell que compleix:

  $n = 2^h - 1$, on n = nombre de nodes i h = altura

3.2. Arbres AVL

DEFINICIONS (II)

- Si T es un arbre binari no buit amb TL i TR com a subarbres esquerre i dret respectivament, llavors T està equilibrat respecte a l'altura si i només si
  - TL i TR són equilibrats respecte a l'altura, i
  - $| hl - hr | \leq 1$ on hl i hr són les altures respectives de TL i TR
- El factor d'equilibri FE (T) d'un node T en un arbre binari es defineix com hr - hl. Per a qualsevol node T en un arbre AVL, es compleix FE (T) = -1, 0, 1



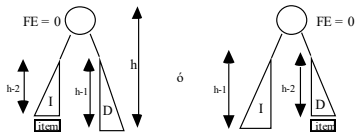
3.2. Arbres AVL

OPERACIONS BÀSIQUES. INSERCIÓ (I)

- Representació d'arbres AVL
  - Mantindre la informació sobre l'equilibri de forma implícita en l'estructura de l'arbre
  - Atribuir a cada node i emmagatzemar-hi el factor d'equilibri de forma explícita

```
TNodoArb {
        Titem fitem;
        TArbBin fiz, fde;
        int FE;   }
```

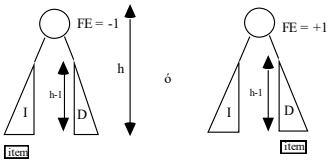
- Inserció en arbres AVL. Casos:
  - Després de la inserció de l'ítem, els subarbres I i D igualaran les seues altures



3.2. Arbres AVL

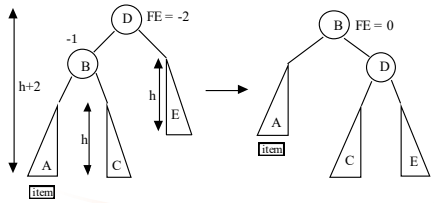
OPERACIONS BÀSIQUES. INSERCIÓ (II)

  - Després de la inserció, I i D tindran distinta altura, però sense vulnerar la condició d'equilibri



- Si hI > hD i es fa inserció en I, o hI < hD i es fa inserció en D.
  Formes de rotació: II, ID, DI, DD

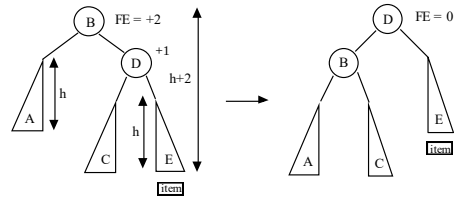
- ROTACIÓ II
  (-2,-1)



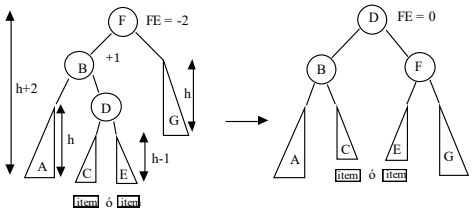
3.2. Arbres AVL

OPERACIONS BÀSIQUES. INSERCIÓ (III)

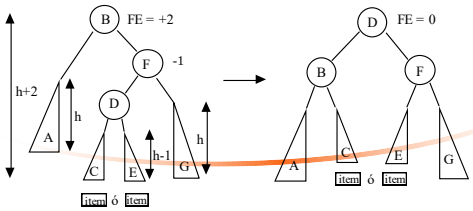
- ROTACIÓ DD
  (+2,+1)



- ROTACIÓ ID
  (-2,+1)



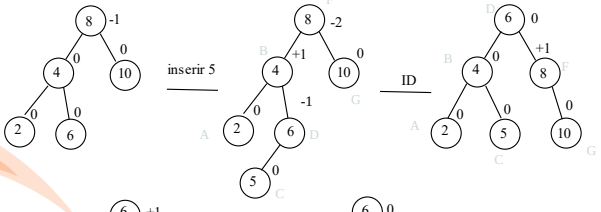
- ROTACIÓ DI
  (+2,-1)

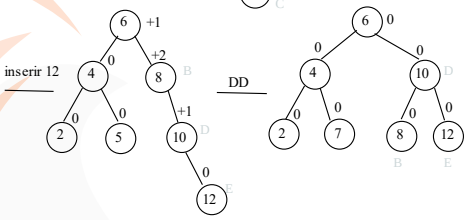


3.2. Arbres AVL

OPERACIONS BÀSIQUES. INSERCIÓ. EXEMPLE (IV)

- **Exemple**. Inserir en el següent arbre els elements 5 i 12





- Cal tindre en compte que l'actualització del FE de cada node s'efectua des les fulles cap a l'arrel de l'arbre

3.2. Arbres AVL

OPERACIONS BÀSIQUES. INSERCIÓ. IMPLEMENTACIÓ (V)

```
ALGORITMO INSERTAR
    ENTRADA/SALIDA
            A: AVL; c : Item
    VAR I : Iterador ; Crece : Integer ;
    METODO
            I = Primer ( A ) ;
            InsertarAux ( I, c, Crece ) ;
    fMETODO
```

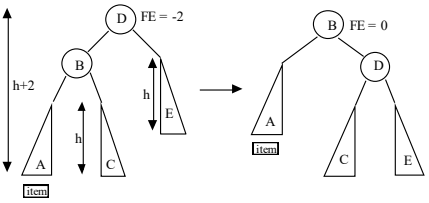
```
ALGORITMO INSERTARAUX
ENTRADA/SALIDA I : Iterador;  Crece: Integer; c : Item ;
VAR CreceIz, CreceDe : Integer ; B : Arbol ;
METODO
si EsVacioArbIt ( I ) entonces
    B = Enraizar ( c ) ; Mover ( I, B ) ; Crece = TRUE ;
sino
    CreceHijo = CreceIz = CreceDe = FALSE ;
    si ( c < Obtener ( I ) ) entonces
        INSERTARAUX ( HijoIzq ( I ), c, CreceIz ) ;
        CreceHijo = CreceIz ;
    sino
        si ( c > Obtener ( I ) ) entonces
            INSERTARAUX ( HijoDer ( I ), c, CreceDe ) ;
            CreceHijo = CreceDe ;
        fsi
    fsi
    si CreceHijo entonces
        caso de:
            1) ( CreceIz y FE ( I ) = 1 ) ó ( CreceDe y FE ( I ) = -1 ) :
    Crece = FALSE ; FE ( I ) = 0 ;
            2) CreceIz y FE ( I ) = 0 : FE ( I ) = -1 ; Crece = TRUE;
            3) CreceDe y FE ( I ) = 0 : FE ( I ) = 1 ; Crece = TRUE;
            4) CreceIz y FE ( I ) = -1 : EquilibrarIzquierda ( I, Crece ) ;
            5 ) CreceDe y FE ( I ) = 1 : EquilibrarDerecha ( I, Crece ) ;
                    fcaso
                sino
                    Crece=FALSE;
    fsi
fsi
fMETODO
```


3.2. Arbres AVL

OPERACIONS BÀSIQUES. INSERCIÓ. IMPLEMENTACIÓ (VI)

```
ALGORITMO EQUILIBRARIZQUIERDA
    ENTRADA/SALIDA I : Iterador;  Crece: Integer;
    VAR J, K: Iterador; int E2;
    METODO
            si ( FE (HijoIzq (I ) = -1 entonces          //ROTACIÓN II
                    Mover (J, HijoIzq (I));
                    Mover (HijoIzq (I), HijoDer (J));
                    Mover (HijoDer (J), I);
                    FE (J) = 0; FE (HijoDer (J)) = 0;
                    Mover (I,J);
            sino                                         //ROTACIÓN ID
                    Mover (J, HijoIzq (I));
                    Mover (K, HijoDer (J));
                    E2 = FE (K);
                    Mover (HijoIzq (I), HijoDer (K));
                    Mover (HijoDer (J), HijoIzq (K));
                    Mover (HijoIzq (K), J);
                    Mover (HijoDer (K), I);
                    FE (K) = 0;

                    caso de E2
                        -1: FE (HijoIzq (K)) = 0; FE (HijoDer (K)) = 1;
                        +1: FE (HijoIzq (K)) = -1; FE (HijoDer (K)) = 0;
                        0: FE (HijoIzq (K)) = 0; FE (HijoDer (K)) = 0;
                    fcaso
                    Mover (I, K);
            fsi
            Crece = FALSE;
    fMETODO
```



3.2. Arbres AVL

EXERCICIS *inserció*

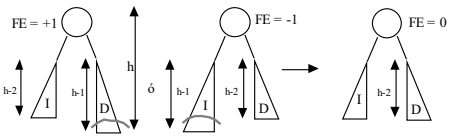
1) Construïu un arbre AVL format pels nodes inserits en el següent ordre amb etiquetes 4, 5, 7, 2, 1, 3, 6

2) Inseriu les mateixes etiquetes amb el següent ordre: 1, 2, 3, 4, 5, 6, 7

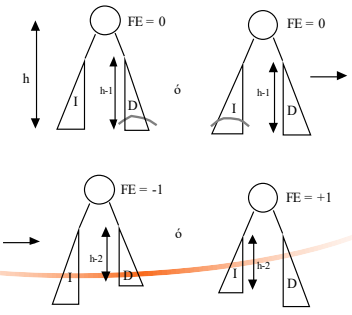
3.2. Arbres AVL

OPERACIONS BÁSIQUES. ESBORRAT (I)

- Esborrat en arbres AVL. Casos:
  - Esborrar l'ítem ens portará en l'arbre a un FE = 0, no serà necessari reequilibrar



- Esborrar l'ítem ens portará en l'arbre a un FE = ±1; en aquest cas tampoc serà necessari reequilibrar

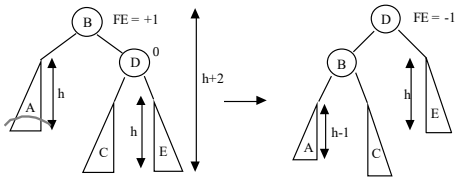


3.2. Arbres AVL

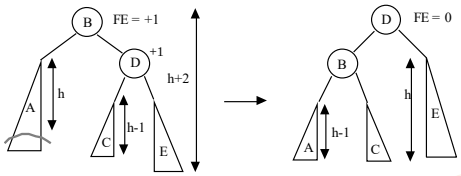
OPERACIONS BÀSIQUES. ESBORRAT (II)

- Rotacions simples

- ROTACIÓ DD
  (+2,0)



(+2,+1)
L'altura de l'arbre decreix

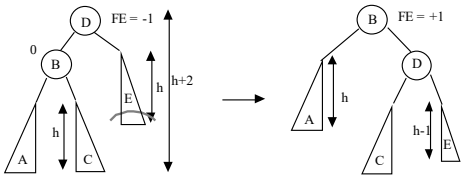


3.2. Arbres AVL

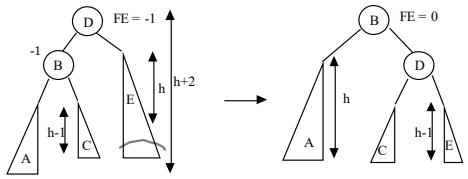
OPERACIONS BÀSIQUES. ESBORRAT (III)

- Rotacions simples

- ROTACIÓ II
  (-2,0)



(-2,-1)
L'altura de l'arbre decreix

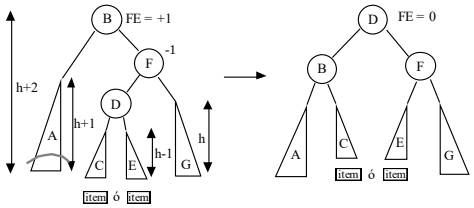


3.2. Arbres AVL

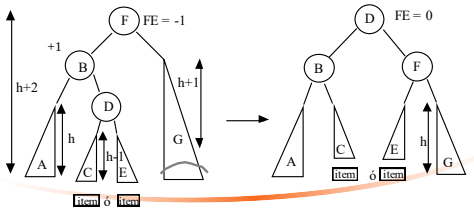
OPERACIONS BÀSIQUES. ESBORRAT (IV)

- Rotacions dobles

- ROTACIÓ DI
  (+2,-1)
  L'altura de l'arbre decreix



- ROTACIÓ ID
  (-2,+1)
  L'altura de l'arbre decreix



3.2. Arbres AVL

OPERACIONS BÀSIQUES. INSERCIÓ I ESBORRAT

- **Estudi de les complexitats d'ambdós algoritmes**
  - L'anàlisi matemàtica de l'algorisme d'inserció és un problema encara no resolt. Els assatjos empírics donen suport la conjectura que l'altura esperada per a l'arbre AVL de *n* nodes és

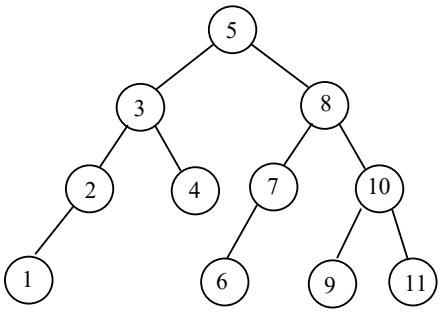
    $$h = \log2 (n) + c \quad / \ c \text{ és una constant xicoteta}$$

  - Aquests arbres han d'utilitzar-se només si les recuperacions d'informació (cerques) són considerablement més freqüents que les insercions → a causa de la complexitat de les operac. d'equilibratge
  - Es pot esborrar un element en un arbre equilibrat amb log (*n*) operacions (en el cas més desfavorable)
- **Diferències operacionals d'esborrat i inserció:**
  - En fer una inserció d'una sola clau es pot produir com a màxim una rotació (de dos o tres nodes)
  - L'esborrat pot requerir una rotació en tots els nodes del camí de cerca
  - Les anàlisis empíriques donen com a resultat que, mentre es presenta una rotació per cada dos insercions,
  - només se'n necessita una per cada cinc esborrats. L'esborrat en arbres equilibrats és, doncs, tan senzill (o tan complicat) com la inserció

3.2. Arbres AVL

EXERCICIS esborrat

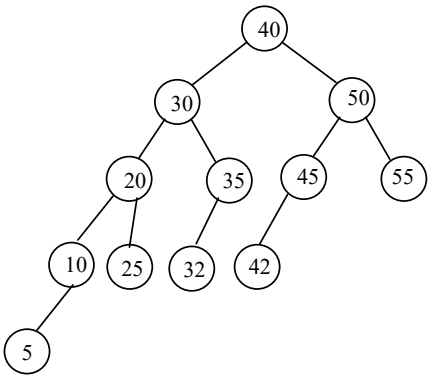
1) Donat el següent arbre AVL d'entrada, feu-hi els esborrats següents mateix: 4, 8, 6, 5, 2, 1, 7. (Nota: en esborrar un node amb 2 fills, substituïu pel major de l'esquerra)



3.2. Arbres AVL

EXERCICIS esborrat

2) Donat el següent arbre AVL d'entrada, feu-hi els esborrats següents: 55, 32, 40, 30. (Nota: en esborrar un node amb 2 fills, substituïu pel major de l'esquerra)



3.2. Arbres AVL

Preguntes de tipus test: Vertader vs. Fals

- Els arbres AVL són aquells on el nombre d'elements en els subarbres esquere i dret difereixen com a molt en 1
- Quan es fa un esborrat en un arbre AVL, en el camí de tornada arrere per actualitzar els factors d'equilibri, com a molt només es fa una rotació
- El següent arbre està equilibrat respecte a l'altura