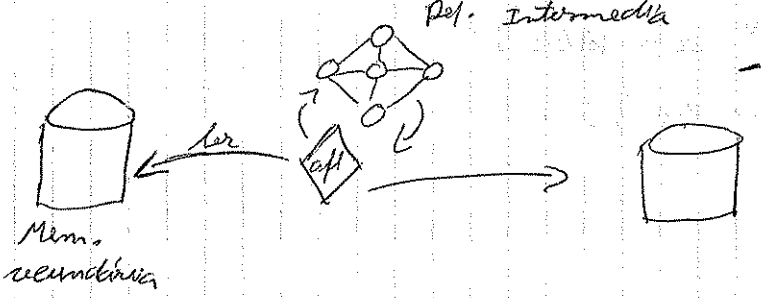
LCC - T - (52)

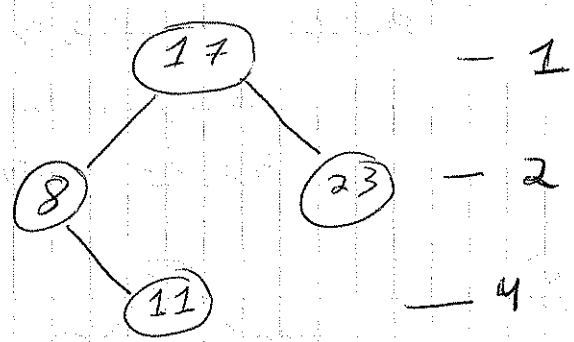
Nelson Brandão, 57872 — 18/05/2010

Árvores Binárias de Procura



- Listas
  - estática – ordem
  - dinâmica

Lista com n elem
n/2 ⇒ log n



níveis = $2^n - 1$ nodos

Haskell

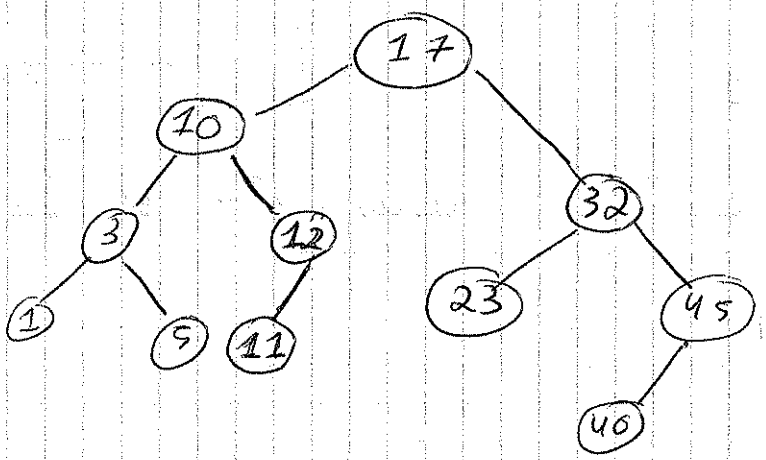
```
data ArvBin a = ConsArvBinNull
              | consArvBin a (ArvBin a) (ArvBin a)
```

C

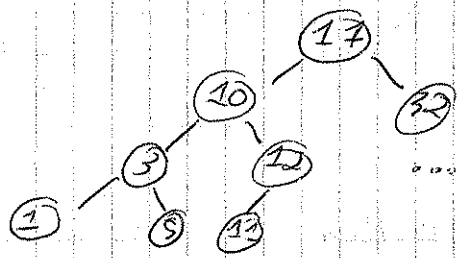
```
typedef struct sArvBin
{
    int valor;
    struct sArvBin *esq, *dir;
} *ArvBin, NodoArvBin;

int main ()
{
    ArvBin a1 = NULL;
    a1 = consArvBin (a1, 17);
    a1 = consArvBin (consArvBin (a1, 32), 8);
    inorder (a1);
} ...
```

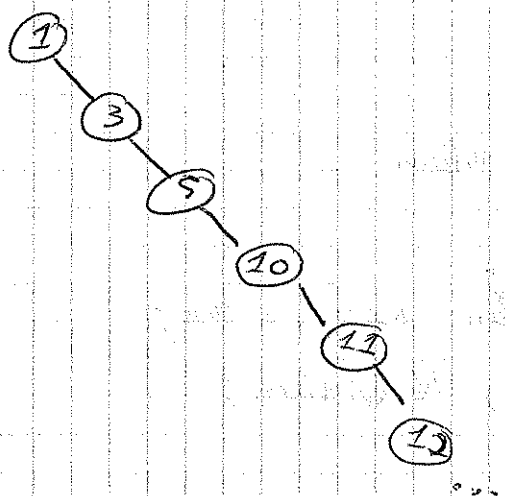
```
void inorder (ArvBin a)
{ if (a)
  { inorder (a -> esq);
    printf ("%d", a -> Valor);
    inorder (a -> dir);
  }
}
```



preorder: 17, 10, 3, 1, 5, 12, 11, 32, 23, 45, 40 - gravar



inorder: 1 3 5 10 11 12 ...



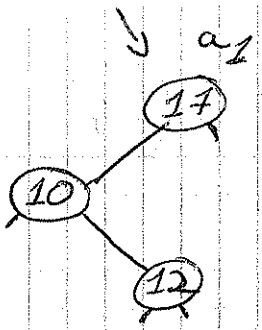
inorder: 1, 5, 3, 11, 12, 10, 23, 40, 45, 32, 17 - destruir

```
void preorder (ArvBin a)
{ if (a)
  { printf (...);
    preorder (a->esq);
    preorder (a->dir);
  }
}

void posorder (ArvBin a)
{ if (a)
  {
    posorder (a->esq);
    posorder (a->dir);
    printf (...);
  }
}

int contaelems (ArvBin a)
{
  if (!a)
  return 0;
  else
    return 1 + contaelems (a->esq) + contaelems (a->dir);
}
```

LCC - T - (55)




```
ArvBin consArvBin (ArvBin a)
{
  ArvBin aux;
  if (!a)
  {
    aux = (ArvBin) malloc (sizeof (NodoArvBin));
    aux -> valor = n;
    aux -> esq = NULL;
    aux -> dir = NULL;
    return aux;
  }
  else if (a -> valor > n)
    { a -> esq = consArvBin (a -> esq, n);
      return a;
    }
  else
    { a -> dir = consArvBin (a -> dir, n);
      return a;
    }
}
```