

Tabelas de símbolos de árvores binárias de busca

▷ Além do Sedgewick (**sempre** leiam o Sedgewick), veja

- <http://www.ime.usp.br/~pf/mac0122-2002/aulas/symbol-table.html>
- <http://www.ime.usp.br/~pf/algoritmos/aulas/bint.html>
- <http://www.ime.usp.br/~pf/algoritmos/aulas/binst.html>

Árvores binárias de busca: rotações

- ▶ *Rotações*: operações importantes, usadas em diversos algoritmos para manipulação de ABBs

Rotações

```
/* prog12.11.c */
```

```
link rotR(link h)
```

```
{ link x = h->l; h->l = x->r; x->r = h;
  return x; }
```

```
link rotL(link h)
```

```
{ link x = h->r; h->r = x->l; x->l = h;
  return x; }
```

Inserção na raiz de uma ABB

```
/* prog12.12.c */
link insertT(link h, Item item)
{ Key v = key(item);
  if (h == z) return NEW(item, z, z, 1);
  if (less(v, key(h->item)))
    { h->l = insertT(h->l, item); h = rotR(h); }
  else
    { h->r = insertT(h->r, item); h = rotL(h); }
  return h;
}

void STinsert(Item item)
{ head = insertT(head, item); }
```

Exercício

Exercício 1. *Adicione o código necessário nas rotinas acima para manter o campo N.*

Tabelas de símbolos

```
/* prog12.1.c - ST.h */  
void STinit(int);  
    int STcount();  
void STinsert(Item);  
Item STsearch(Key);  
void STdelete(Item);  
Item STselect(int);  
void STsort(void (*visit)(Item));
```

Implementação das demais funções para ABBs

▷ STselect()

▷ STdelete()

Seleção com uma ABB

```
/* prog12.13.c */
Item selectR(link h, int k)
{ int t = h->l->N;
  if (h == z) return NULLitem;
  if (t > k) return selectR(h->l, k);
  if (t < k) return selectR(h->r, k-t-1);
  return h->item;
}

Item STselect(int k)
{ return selectR(head, k); }
```


Partição de uma ABB

```
/* prog12.14.c */
link partR(link h, int k)
{ int t = h->l->N;
  if (t > k )
    { h->l = partR(h->l, k); h = rotR(h); }
  if (t < k )
    { h->r = partR(h->r, k-t-1); h = rotL(h); }
  return h;
}
```

Remoção de uma ABB

```
/* prog12.15.c */
link joinLR(link a, link b)
{
    if (b == z) return a;
    b = partR(b, 0); b->l = a;
    return b;
}
```

Remoção de uma ABB

```
/* prog12.15.c */
[...]
```

```
link deleteR(link h, Key v)
{ link x; Key t = key(h->item);
  if (h == z) return z;
  if (less(v, t)) h->l = deleteR(h->l, v);
  if (less(t, v)) h->r = deleteR(h->r, v);
  if (eq(v, t))
    { x = h; h = joinLR(h->l, h->r); free(x); }
  return h;
}
```

```
void STdelete(Key v)
{ head = deleteR(head, v); }
```

União de TSs/ABBs

- ▷ STjoin()
- ▷ Algoritmo trivial (seqüência de inserções)
- ▷ Algoritmo linear para ABBs

União de ABBs

```
/* prog12.16.c */
link STjoin(link a, link b)
{
    if (b == z) return a;
    if (a == z) return b;
    b = STinsert(b, a->item); /* insercao na raiz */
    b->l = STjoin(a->l, b->l);
    b->r = STjoin(a->r, b->r);
    free(a);
    return b;
}
```