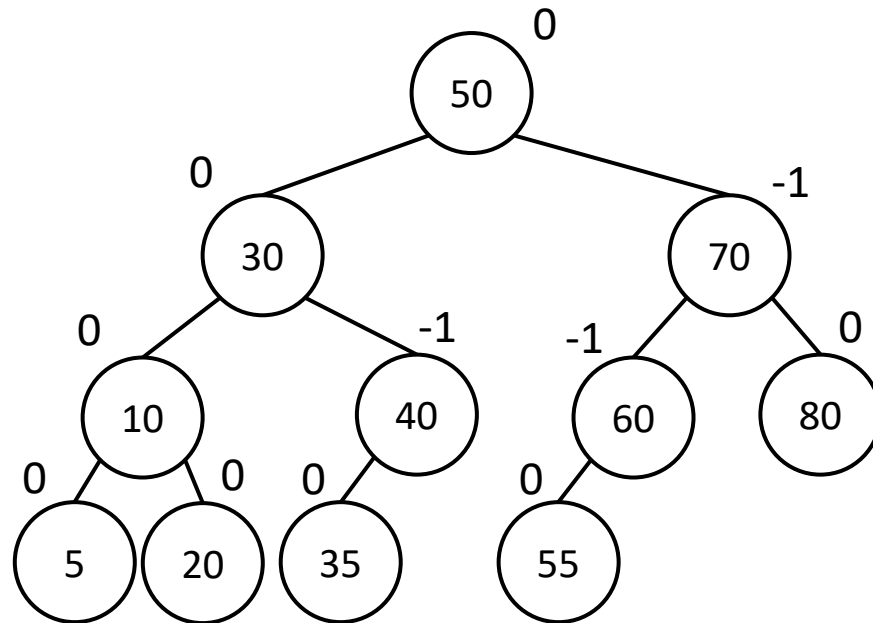


# Estruturas de Dados Avançadas (INF1010) 3WB

## Lab4: Inserção em Árvores AVL

# Árvore AVL



# Inserção em AVL

```
Avl* avl_insere2(Avl* r, int chave, tdados *dados, int* cresceu) {
    if (r==NULL) {
        r = (Avl*) malloc(sizeof(Avl));
        r->esq = r->dir = NULL;
        r->chave = chave;
        r->dados = dados;
        r->fb = 0;
        *cresceu = 1;
    }
    else if (r->chave > chave) {
        r->esq = avl_insere2 (r->esq, chave, dados, cresceu);
        if (*cresceu) ...
    }
    else if (r->chave < chave) {
        r->dir = avl_insere2 (r->dir, chave, dados, cresceu);
        if (*cresceu) ...
    }
    return r;
}

Avl* avl_insere(Avl* r, int chave, tdados *dados) {
    int cresceu;
    return avl_insere2 (r, chave, dados, &cresceu);
}
```

# Inserção em AVL

```
if (*cresceu) {    /* r cresceu à esquerda (ou seja, he aumentou)*/  
  
    switch(r->fb) { /* análise do fator de balanceamento de r */  
  
        case 1: /* antes: hd>he - depois: hd=he pois he aumentou */  
            r->fb = 0; *cresceu = 0; break;  
  
        case 0: /* antes: hd=he - depois: hd<he pois he aumentou */  
            r->fb = -1; break; /* avisa para cima que cresceu */  
  
        case -1: /* antes: hd<he - depois: hd-he=-2 pois he aumentou */  
            if (r->esq->fb == -1) r = avl_rotacao_direita(r);  
            else r = avl_rotacao_esquerda_direita(r);  
            *cresceu = 0; break;  
        }  
    }
```

# Lab de hoje

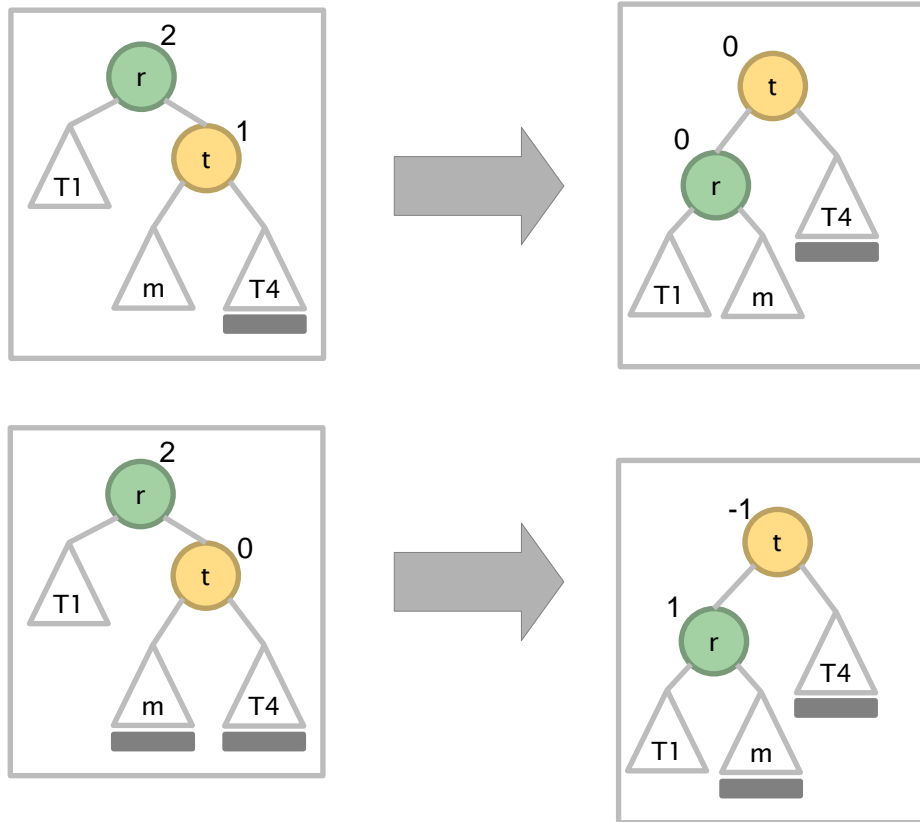
1. Terminar de implementar as rotações

2. Implementar inserção em árvore AVL

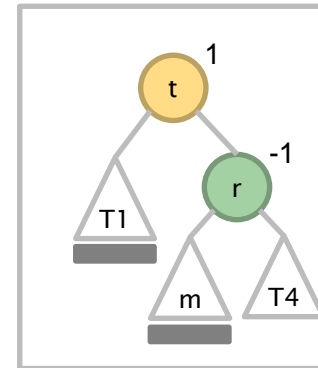
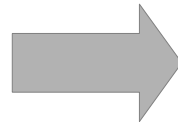
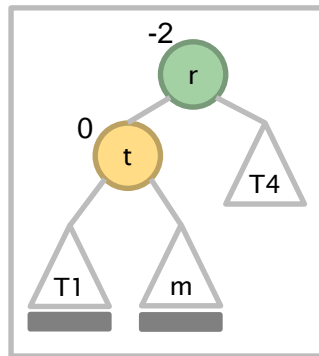
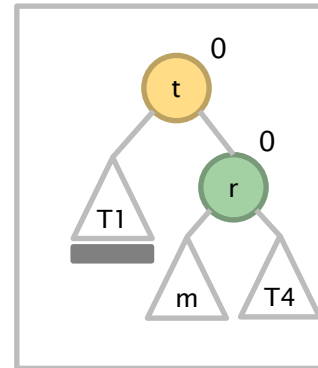
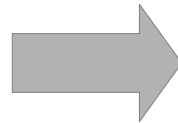
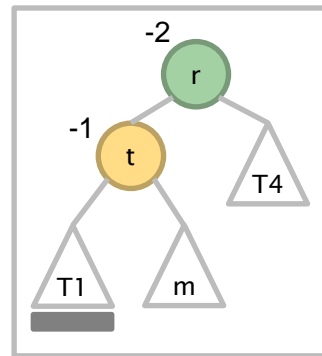
- Visualização de árvore AVL

`https://www.cs.usfca.edu/~galles/visualization/AVLtree.html`

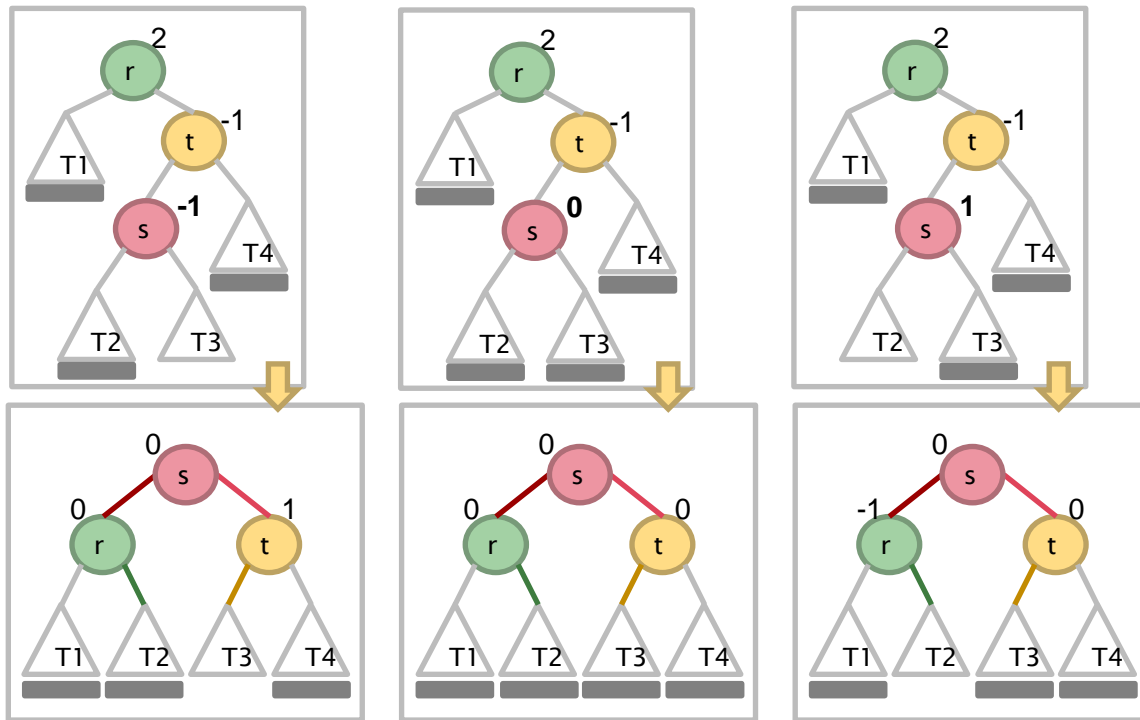
# Resumo: Rotação Simples Esquerda



# Resumo: Rotação Simples Direita



# Resumo: Rotação Dupla Direita-Esquerda





# Resumo: Rotação Dupla Esquerda-Direita

