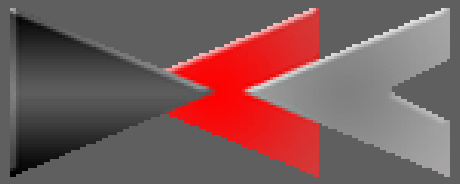
DCC107



Laboratório de Programação II

Tipos Abstratos de Dados Árvores Binárias e Árvore Binária de Busca
Exercícios em C

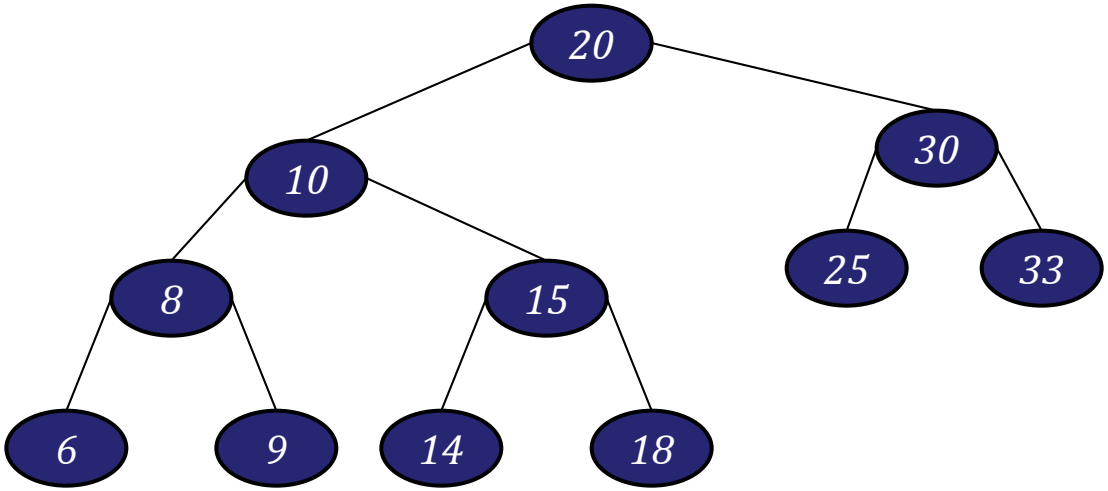
Árvore Binária

- Usando o TAD criado na 1ª aula e a função main abaixo:

```
int main()
{
    Arv *no1, *no2, *arv;
    no1 = cria(6,NULL,NULL);
    no2 = cria(9,NULL,NULL);

    arv = cria(8, no1, no2);

    no1 = cria(14,NULL,NULL);
    no2 = cria(18,NULL,NULL);

    arv = cria(10, arv, cria(15, no1, no2));

    no1 = cria(25,NULL,NULL);
    no2 = cria(33,NULL,NULL);

    arv = cria(20, arv, cria(30, no1, no2));

    imprime(arv);
    libera(arv);
    return 0;
}
```

Árvore Binária

- Cria a seguinte árvore:



- Comando `imprime(arv)`: 20 10 8 6 9 15 14 18 30 25 33 (in ordem)

Árvore Binária

- Exercício 1: fazer três funções em C para calcular e imprimir a média dos valores dos nós da árvore criada anteriormente. Para cada função realizar:
  - percurso in ordem (central).
  - percurso em pré ordem.
  - percurso em pós ordem.

Media = SomaNos/NumeroNos

Para a solução do problema, serão feitas uma função para calcular a soma dos valores dos nós e outra para calcular o número de nós. Uma terceira calculará, enfim, a média dos valores dos nós da árvore binária.

Árvore Binária

- Soma: percurso in ordem (central)

```
float somanos(Arv *a)
{
    if(vazia(a))
        return 0;
    else
        return somanos(a->esq) + a->info + somanos(a->dir);
}
```

- Soma: percurso em pré ordem

```
float somanos(Arv *a)
{
    if(vazia(a))
        return 0;
    else
        return a->info + somanos(a->esq) + somanos(a->dir);
}
```

- Soma: percurso em pós ordem

```
float somanos(Arv *a)
{
    if(vazia(a))
        return 0;
    else
        return somanos(a->esq) + somanos(a->dir) + a->info;
}
```

Árvore Binária

- Conta número de nós: percurso in ordem (central)

```
int contanos(Arv *a)
{
    if(vazia(a))
        return 0;
    else
        return contanos(a->esq) + 1 + contanos(a->dir);
}
```

- Conta número de nós: percurso em pré ordem

```
int contanos(Arv *a)
{
    if(vazia(a))
        return 0;
    else
        return 1 + contanos(a->esq) + contanos(a->dir);
}
```

- Conta número de nós: percurso em pós ordem

```
int contanos(Arv *a)
{
    if(vazia(a))
        return 0;
    else
        return contanos(a->esq) + contanos(a->dir) + 1;
}
```

Árvore Binária

- Media (fazendo qualquer um dos percursos)

```
float media(Arv *a)
{
    if(!vazia(a))
        return somanos(a)/contanos(a);
}
```

- Observe que as funções `contanos` e `somanos` não são operações do TAD `Arv`, sendo assim elas não devem ser incluídas no arquivo `Arvbin.h` (usado apenas neste arquivo).
- Há um problema com esta abordagem: a árvore é percorrida duas vezes, uma para calcular a soma e outra para contar o número de nós.
- A seguir será feito um procedimento para calcular média realizando-se apena um percurso na árvore binária.

Árvore Binária

- Procedimento para o cálculo da média:

```
float medianos2(Arv *a)
{
    int num_nos = 0;      float soma = 0.0;
    if(!vazia(a))
    {
        conta_soma(a, &soma, &num_nos);
        return soma/num_nos;
    }
    else
        return 0.0;
}
```

- A função `medianos2` é uma operação do TAD árvore, portanto ele deve ser incluída no arquivo `Arvbin.h`.
- As variáveis `soma` e `num_nos` são passadas por referência à função `conta_soma` que calcula, respectivamente, nas variáveis `num_nos` e `soma`, o número de nós da árvore e soma dos valores dos nós.

Árvore Binária

- No procedimento a seguir é necessário passar os valores a serem calculados por referência (ponteiro). Conta o número de nós e soma os seus valores fazendo um percurso em **pré ordem**.

```
void conta_soma(Arv *a, float *soma, int *num_nos)
{
    if(!vazia(a))
    {
        *soma = a->info + *soma;
        *num_nos =  *num_nos + 1;
        conta_soma(a->esq, soma, num_nos);
        conta_soma(a->dir, soma, num_nos);
    }
}
```

- Observe que este procedimento não é uma operação do TAD `Arv`, sendo assim ele não deve ser incluído no arquivo `Arvbin.h` (usado apenas neste arquivo).

Árvore Binária

- Conta o número de nós e soma os seus valores fazendo um percurso em **in ordem**.

```
void conta_soma(Arv *a, float *soma, int *num_nos)
{
    if(!vazia(a))
    {
        conta_soma(a->esq, soma, num_nos);
        *soma = a->info + *soma;
        *num_nos =  *num_nos + 1;
        conta_soma(a->dir, soma, num_nos);
    }
}
```

Árvore Binária

- Conta o número de nós e soma os seus valores fazendo um percurso em **pós ordem.**

```
void conta_soma(Arv *a, float *soma, int *num_nos)
{
    if(!vazia(a))
    {
        conta_soma(a->esq, soma, num_nos);
        conta_soma(a->dir, soma, num_nos);
        *soma = a->info + *soma;
        *num_nos =  *num_nos + 1;
    }
}
```

Árvore Binária

- Função `main()` abaixo

```
int main()
{
     ...... //igual a main() anterior
   printf("\n%f\n", medianos(arv));
   printf("\n%f\n", medianos2(arv));
   libera(arv);
   return 0;
}
```

- Será impresso:

```
17.090910
17.090910
```

Árvore Binária

- Exercício 2: fazer três procedimentos em C para imprimir todos os valores dos nós de um dado nível `k`. Para cada função realizar:
  - percurso in ordem (central).
  - percurso em pré ordem.
  - percurso em pós ordem.

- Neste caso, o resultado deverá ser diferente?

Árvore Binária

- Função `imprimir_nivel`. Usada apenas para inicializar o valor corrente do nível do nó (valor 0):

```
void imprimir_nivel(Arv *a, int k)
{
    imp_nivel(a, 0, k);
}
```

- A função abaixo, implementa recursivamente a impressão dos valores do k-ésimo nível da árvore em pré ordem.

```
void imp_nivel(Arv *a, int cor_n, int k)
{ //cor_n: valor (inteiro) corrente do nível do nó visitado.
    if(!vazia(a))
    {
        if(cor_n == k)
            printf("nivel: %d,  valor: %d\n",cor_n, a->info);
        imp_nivel(a->esq, cor_n + 1, k);
        imp_nivel(a->dir, cor_n + 1, k);
    }

}
```

Árvore Binária

- Considerando a árvore exemplo, será impresso:

```
nivel: 2, valor: 8
nivel: 2, valor: 15
nivel: 2, valor: 25
nivel: 2, valor: 33
```

- em in ordem.

```
void imp_nivel(Arv *a, int cor_n, int k)
{ //cor_n: valor (inteiro) corrente do nível do nó.
    if(!vazia(a))
    {
        imp_nivel(a->esq, cor_n + 1, k);
         if(cor_n == k)
            printf("nivel: %d,  valor: %d\n",cor_n, a->info);
        imp_nivel(a->dir, cor_n + 1, k);
    }

}
```

Árvore Binária

- Considerando a árvore exemplo, será impresso:

```
nivel: 2, valor: 8
nivel: 2, valor: 15
nivel: 2, valor: 25
nivel: 2, valor: 33
```

Árvore Binária

- Exercícios:
  - Fazer uma função (ou procedimento) para calcular a média dos valores dos nós de um nível *k* dado.
  - Através de percursos in ordem, pós ordem e pré ordem, fazer funções para calcular a quantidade de valores ímpares armazenados nos nós folhas de uma árvore binária.
  - Implementar uma função para calcular e retornar a quantidade de valores ímpares numa árvore binária qualquer (não é binária de busca). A árvore deve ser passada como parâmetro.

Árvore Binária de Busca

- Exercício 3: Considerando uma árvore binária de busca de números inteiros, fazer um função para calcular e retornar a média dos valores dos nós que formam o caminho percorrido para localizar, com sucesso ou não, o nó que tem valor igual a chave CH dada. O nó com valor CH, se houver, entra no cálculo da média. Definir os tipos utilizados no procedimento.

Árvore Binária de Busca

- Caminhamento em árvore binária de busca que soma os valores dos nós e conta a quantidade deles atingidas no percurso. Depois calcula a média

```
float media_arv_bb(Arv *a, int ch)
{
    Arv * no = a;    float soma = 0.0;    int numnos = 0;

    while(no != NULL && no->info != ch){
        soma += no->info;
        numnos++;
        if(ch > no->info)
            no = no->dir;
        else
            no = no->esq;
    }
    if(!vazia(no))
    {  //soma e conta o nó com valor ch
        soma += no->info;
        numnos++;
    }
    return soma/numnos;
}
```