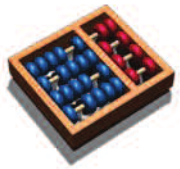


Instituto de Computação – UNICAMP
MC202 – Estrutura de Dados
Prof.: Neucimar J. Leite, Orlando Lee

Laboratório Nº 7

O objetivo deste laboratório é implementar uma árvore B, conforme visto em aula, para guardar RAs e nomes de alunos.

O programa deve implementar inserção e consulta usando-se os RAs dos alunos como chave. Ele deve também permitir a impressão a árvore com todos RAs.

Entrada

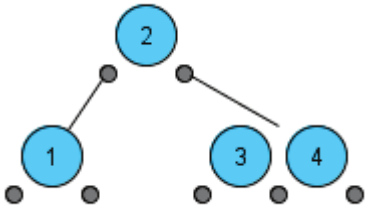
A entrada do programa recebe os seguintes parâmetros, em ordem:

- **Ordem mínima L**: inteiro maior ou igual a 2 que representa a quantidade mínima de filhos que cada nó interno da árvore pode ter.
- **Número de alunos** a serem **inseridos** na árvore.
- **Lista com RAs e nomes** dos alunos **inseridos**.
- **Número de alunos** a serem **consultados** na árvore.
- **Lista com RAs** dos alunos **consultados**.

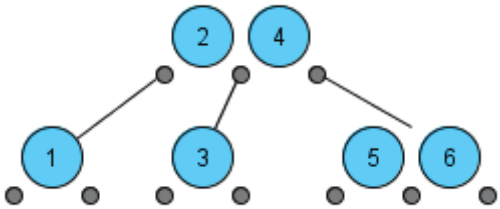
A ordem mínima L para esse lab, que não implementa remoção, não é significativa por si, mas serve de base para o cálculo da ordem máxima U = 2L, que representa a quantidade máxima de filhos que cada nó da árvore pode ter. Assim, cada nó tem de L a 2L filhos, ou de (L – 1) a (2L – 1) chaves.

Considere a seguir um exemplo de inserções de chaves em árvore B com L = 2, o que implica em U = 4, ou seja, máximo de 4 filhos (3 chaves) por nó. Nesse exemplo vamos inserir as chaves 1 a 10, nessa ordem. Como podemos ver na figura 1, até a inserção da chave 3, não é necessário dividir o nó raiz, pois ele pode ter até 3 chaves.

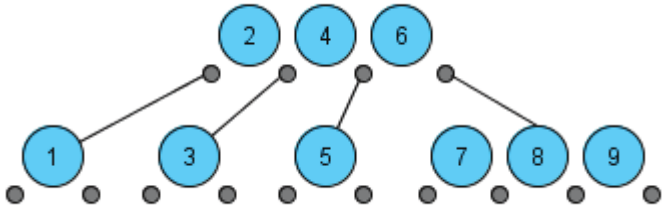
Entretanto, ao inserir a chave 4, é necesário quebrar o nó raiz e promover uma chave. Quando isso ocorrer, sempre a chave do meio, antes da inserção, é promovida (perceba que o número máximo de chaves por nó é sempre ímpar (2L – 1), não havendo ambiguidade ao escolher a chave do meio). Assim a árvore B fica com a configuração da figura 2.



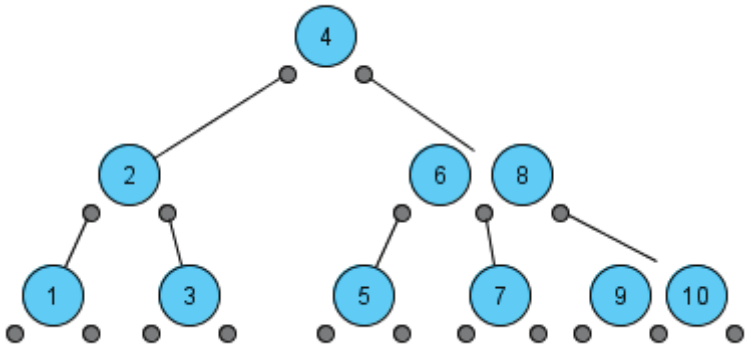
A chave 5 então é inserida no mesmo nó das chaves 3 e 4, e ao inserir a chave 6, a chave 4, que é a do meio, é promovida, dividindo o mesmo nó que a chave 2, como mostra a figura 3.



Analogamente, o mesmo é feito até a inserção da chave 9 (figura 4).



Nesse ponto, a inserção da chave 10 causa a promoção da chave 8. Se necessário, a promoção deve ser efetuada recursivamente até a raiz. Nesse caso, a chave 4 é também promovida. A configuração final da árvore B desse exemplo está ilustrada na figura 5.



Saída

A saída do programa deve imprimir os nomes correspondentes aos RAs dos alunos consultados (um nome por linha, não imprime alunos não encontrados). A busca pelos alunos deve ser feita individualmente, em profundidade na árvore. Se o parâmetro de entrada referente ao número de alunos a serem consultados for zero, a saída deve imprimir a árvore com todos RAs (percurso em largura), com o seguinte formato:

{ *primeiro_ra_da_raiz_segundo_ra_da_raiz ... último_ra_da_raiz* }
{ *ras_primeiro_no_nível2* }{ *ras_segundo_no_nível2* }...{ *ras_último_no_nível2* }
.
.
.
{ *ras_primeiro_no_folha* }{ *ras_segundo_no_folha* }...{ *ras_último_no_folha* }

Os exemplos dos testes estão na página desta atividade no Susy.

Entrega

`arvoreb.h`: assinaturas de funções e definições de estruturas que implementam árvore B.
`arvoreb.c`: implementa funções definidas em `arvoreb.h`, além de funções auxiliares necessárias.
`main.c`: contém a função main, que lê os parâmetros de entrada e chama as funções de `arvoreb.h`. Esse arquivo é previamente fornecido e não deve ser modificado.

Download do arquivo arvoreb.h em <http://www.students.ic.unicamp.br/~ra116199/arvoreb.h>

Avaliação

A nota deste laboratório será dada de acordo com a coerência com o enunciado e da documentação fornecida (comentários no código).