Trabalho 1: Árvore binária de busca

Casos de Teste

Este documento apresenta instruções para a implementação dos casos de teste do primeiro trabalho da disciplina INF1010 turma 3WB 2018.1:

Uma aplicação de uma árvore binária de busca: Dicionário de textos (strings).

Instruções para a implementação dos testes foram dadas em sala de aula e podem também ser encontradas na página inicial da plataforma de testes.

Dica: Para testar a sua implementação localmente, direcionando um arquivo para a entrada padrão, adicione por exemplo "`< arquivo_entrada.txt`" ao comando de executar. Isso pode ser feito no terminal, ou configurado nas IDEs. Da mesma forma, se quiser direcionar a saída para um arquivo, utilize "`> arquivo_saida.txt`".

Caso 1: Inserir e imprimir (2,5pts)

Entrada:

- **Uma string** indicando o caso de teste.
- **Um inteiro** indicando a quantidade de testes que serão feitos (ou árvores que serão criadas).
- Para cada teste:
 - **um inteiro n** indicando a quantidade de itens que serão inseridos na árvore.
 - **uma sequência de n linhas**, compostas por **duas strings** separadas por espaço, que são a chave e o valor, respectivamente.
ATENÇÃO: As strings utilizadas para teste NÃO contém espaço ou qualquer outro caractere em branco.

Saída: Para cada árvore criada, após inseridos todos os seus elementos, ela deve ser impressa:

- em formato de árvore (padrão <> da função mostraAbb())
 - em ordem lexicográfica das suas chaves (função mostra())
- Uma linha deve ser saltada entre cada árvore impressa.

Exemplo de entrada e saída:

Entrada	Saída
TESTCASE1 2 10 j ydo f nte a oqt c cvg g vvz b uqv e vja d brn h aay i qfq 5 H3V SiMvAyVE kus Y98rPfbN nDA jdtvIFdf iKL 4fLdJPoK nyN XiYR93E5	<j<f<a.<c<b...><e<d...>.>><g.<h.<i...>>>.> (a,b,c,d,e,f,g,h,i,j,) <H3V.<kus<iKL...><nDA.<nyN...>>>> (H3V,iKL,kus,nDA,nyN,)

Caso 2: Obter valores (1,5pts)

Entrada:

- **Uma string** indicando o caso de teste.
- **Um inteiro** indicando a quantidade de testes que serão feitos (ou árvores que serão criadas).
- Para cada teste:
 - **um inteiro n** indicando a quantidade de itens que serão inseridos na árvore.

- **uma sequência de n linhas**, compostas por **duas strings** separadas por espaço, que são a chave e o valor, respectivamente.
- ATENÇÃO:** As strings utilizadas para teste **NÃO** contém espaço ou qualquer outro caractere em branco.
- **um inteiro m** indicando a quantidade de itens que serão consultados na árvore.
- **uma sequência de m linhas**, compostas por **uma string**, que representa a chave que será consultada no dicionário.

Saída:

- Para cada árvore lida:
 - Para cada chave, dentre as m chaves lidas, deve ser impressa uma linha, contendo o valor (string) correspondente consultado na árvore com `obtemValor()`. Pode haver casos de consulta de valor inexistente na árvore.
 - Ao final das m consultas, uma linha deve ser pulada.

Exemplo de entrada e saída:

Entrada	Saída
TESTCASE2	vja
2	
6	cvg
f nte	oqt
a oqt	
c cvg	jdtvIFdf
b uqv	
e vja	
d brn	
4	
e	
j	
c	
a	
5	
H3V SiMvAyVE	
kus Y98rPfbN	
nDA jdtvIFdf	
iKL 4fLdJPoK	
nyN XiYR93E5	
1	
nDA	

Caso 3: Remover (2,5pts)

Entrada:

- **Uma string** indicando o caso de teste.
- **Um inteiro** indicando a quantidade de testes que serão feitos (ou árvores que serão criadas).
- Para cada teste:
 - **um inteiro n** indicando a quantidade de operações (inserção ou remoção) que serão feitas na árvore.
 - **uma sequência de n linhas**, compostas por:
 - * **um inteiro b**, que pode ser 0 ou 1: 0 para remoção, 1 para inserção.
 - * **uma string** representando a chave a ser removida, quando $b = 0$, **OU duas strings** representando chave e valor a serem inseridos, quando $b = 1$. Pode haver casos de tentativa de remoção de um elemento que não existe.

ATENÇÃO: As strings utilizadas para teste **NÃO** contém espaço ou qualquer outro caractere em branco.

Saída: Para cada árvore criada, após todas as m operações, ela deve ser impressa em formato de árvore (padrão $\langle \rangle$ da função `mostraAbb()`). Pule uma linha entre cada árvore impressa.

Exemplo de entrada e saída:

Entrada	Saída
<pre> TESTCASE3 2 10 1 f nte 1 a oqt 1 c cvg 0 a 0 z 1 b uqv 1 e vja 0 f 1 d brn 0 f 5 0 j 1 H3V SiMvAyVE 1 kus Y98rPfbN 1 nDA jdtvIFdf 0 H3V </pre>	<pre> <c<b..><e<d...>.>> <kus.<nDA...>> </pre>

Caso 4: Verificar propriedades da árvore (1,5pts)

Entrada:

- Uma **string** indicando o caso de teste.
 - Um **inteiro** indicando a quantidade de testes que serão feitos (ou árvores que serão criadas).
 - Para cada teste:
 - um **inteiro n** indicando a quantidade de itens que serão inseridos na árvore.
 - uma **sequência de n linhas**, compostas por **duas strings** separadas por espaço, que são a chave e o valor, respectivamente.
- ATENÇÃO:** As strings utilizadas para teste **NÃO** contém espaço ou qualquer outro caractere em branco.

Saída:

- Para cada árvore lida:

- Imprima em uma linha dois inteiros: o **tamanho** e a **altura** da árvore, separados por um espaço.
- Pule uma linha entre cada impressão.

Exemplo de entrada e saída:

Entrada	Saída
TESTCASE4	10 5
2	
10	5 3
j ydo	
f nte	
a oqt	
c cvg	
g vvz	
b uqv	
e vja	
d brn	
h aay	
i qfq	
5	
H3V SiMvAyVE	
kus Y98rPfbN	
nDA jdtvIFdf	
iKL 4fLdJPoK	
nyN XiYR93E5	

Caso 5: Copiar (2pts)

Entrada:

- **Uma string** indicando o caso de teste.
- **Um inteiro** indicando a quantidade de testes que serão feitos (ou árvores que serão criadas).
- Para cada teste:
 - **um inteiro n** indicando a quantidade de itens que serão inseridos na árvore.

- **uma sequência de n linhas**, compostas por **duas strings** separadas por espaço, que são a chave e o valor, respectivamente.

ATENÇÃO: As strings utilizadas para teste **NÃO** contém espaço ou qualquer outro caractere em branco.

Saída:

- Para cada árvore lida:
 - Copie a árvore utilizando o construtor de cópia, e imprima a árvore copiada utilizando a função mostraAbb().
 - Ao final de cada árvore impressa, uma linha deve ser pulada.

Exemplo de entrada e saída:

Entrada	Saída
TESTCASE5 2 10 j ydo f nte a oqt c cvg g vvz b uqv e vja d brn h aay i qfq 5 H3V SiMvAyVE kus Y98rPfbN nDA jdtvIFdf iKL 4fLdJPoK nyN XiYR93E5	<j<f<a.<c<b...><e<d...>.>>><g.<h.<i...>>>.> <H3V.<kus<iKL...><nDA.<nyN...>>>>