UNIVERSIDADE
FEDERAL DO CEARÁ
UFC-CAMPUS CRATEÚS

Science Computer

DATAS STRUCTURE ADVANCED-CRT0026

Exercises And Resumes

Discente
Victor Gabriel Martins de Oliveira Loiola
Matrícula 473091

Professor
Msc. Luiz Alberto do Carmo Viana

Crateús
2020

Sumário

1 Exercises Resolved

1.1 Binary Search Tree

Exercise 1. O que aconteceria caso remove(leftMaxkey); fosse posta como a última instrução em seu bloco?

Prova
Se deslacamos remove(leftMaxKey) como última instrução do seu bloco resultaria em uma **falha de segmentação**, além do mais estaria também duplicando as chave e valor do leftMaxKey.Assim o conteúdo e a chave do **nó** que deseja-se apagar tem chave e valor do leftMaxKey.

Exercise 2. Por que o nó de leftMaxkey tem ao menos uma sub-árvore vazia?

Prova
Já que leftMaxKey é a chave com maior valor das sub-árvores da esquerda do nó pai, mas perceba que leftMaxKey está restrito somente a ter filho esquerdo ou não,resultando uma sub-árvore vazia que nesse caso seria a da direita.Logo concluir-se que leftMaxKey não deve ter filhos com chaves maiores que a sua, pois se tivesse ele não seria leftMaxKey e sim alguma chave maior na sub-árvore a sua direita.
Portanto leftMaxKey se tiver filho só pode ter-lo na sub-árvore a esquerda, o que resta uma sub-árvore vazia, mas no caso de não possuí filhos terá duas sub-árvores vazias.

Exercise 3. Pesquise sobre move semantics em C++.

Prova
TODO escreve isso quando ou transcrever do caderno ou pesquisar.

Exercise 4. Prove que, se um nó em uma Árvore Binária de Busca tem dois filhos, então seu sucessor não tem filho esquerdo e seu antecessor não tem filho direito.

Prova
TODO escreve isso quando ou transcrever do caderno ou pesquisar.

TODO escreve o restante das questões quando termina-las ou transcrever do caderno.

1.2 AVL Tree

Exercise 5. Prove que uma árvore binária de altura h tem no máximo $2^{h+1} - 1$ nós. Dica: tente usar indução.

Prova via indução

Caso Base:
Com altura 0 tempos 1 nó, com altura 1 temos 2 nós e com altura 2 tempos 7 nós.

$$h = 0 \text{ contém 1 nó}$$
$$h = 1 \text{ contém 3 nós}$$
$$h = 2 \text{ contém 7 nós}$$

Agora utilizando a fórmula $2^{h+1} - 1$ para verificarmos o **primeiro caso onde a altura é 0**, como será utilizada o **primeiro princípio da indução**, vulgo indução fraca, verificar até a altura 2 é opcional.

$$\text{com } h = 0$$
$$\text{pela fórmula } 2^{h+1} - 1 \text{ obtemos: } 2^{0+1} - 1 = 1 \text{ nó}$$
$$\text{com } h = 1$$
$$\text{pela fórmula } 2^{h+1} - 1 \text{ obtemos: } 2^{1+1} - 1 = 3 \text{ nós}$$
$$\text{pela fórmula } 2^{h+1} - 1 \text{ obtemos: } 2^{2+1} - 1 = 7 \text{ nós}$$

Como o caso base é verdadeiro, partiremos para o passo indutivo.
Passo Indutivo: $P(1) \rightarrow (h+1)$.
Temos que a fórmula $2^{h+1}-1$ funciona até 2^h, com $h \geq 0$, sendo assim a hipótese de indução($H.I$).Deseja-ja provar que a fórmula vale para os $h+1$-ésimos termos.observe a estrutura abaixo:

$$2^0 + 2^1 + 2^2 + ... + 2^h + 2^{h+1}$$

Perceba que fórmula válida para calcular a quantidade de nós máximos em uma árvore de até 2^h que dá liberdade suficiente para utiliza-se a hipótese de indução($H.I$).
Substituindo $H.I$ no somatório e desenvolvendo a equação após a inserção:

$$
\begin{aligned}
&2^0 + 2^1 + 2^2 + ... + 2^h + 2^{h+1} \\
&= 2^{h+1} - 1 + 2^{h+1} \\
&= 2^{h+1} + 2^{h+1} - 1 \\
&= 2^h(2^1 + 2^1) - 1 \\
&= 2^h \cdot 4 - 1 \\
&= 2^h \cdot 2^2 - 1 \\
&= 2^{h+2} - 1
\end{aligned}
$$

Como concluímos os passos base e passo indutivo, sendo ambos verdadeiros, concluímos que $2^{h+1} - 1$ com h$\geq$ 0 é capaz de resultar no número de nós máximo em uma **árvore binária de busca** com altura h.

TODO escreve o restante das questões quando termina-las ou transcrever do caderno.

1.3 Red-Black Tree

TODO escreve o restante das questões quando termina-las ou transcrever do caderno.

2 Resumes

TODO escreve isso quando elaborá uma certa quantidade X de pequenos resumos.