MAC0323 ALGORITMOS E ESTRUTURAS DE DADOS II

FOLHA DE SOLUÇÃO

Nome: Leonardo Heidi Almeida Murakami **NUSP:** 11260186

Assinatura

Sua assinatura atesta a autenticidade e originalidade de seu trabalho e que você se compromete a seguir o código de ética da USP em suas atividades acadêmicas, incluindo esta atividade.

Exercício: Exercício Teórico I - MAC0323

SOLUÇÃO

1. Exercício 1

Seja T uma Árvore Binária Completa (ABC) com N nós, onde $N \geq 1$. Seja $N = (b_n b_{n-1} \dots b_0)_2$ a representação binária de N, onde $b_n = 1$ (e portanto, $n = \lfloor \lg N \rfloor$). Sejam T_1 e T_2 as subárvores esquerda e direita da raiz de T, com N_1 e N_2 nós, respectivamente.
Queremos provar que se $b_{n-1} = 1$, então $N_1 = 2^n - 1$ e $N_2 = (b_{n-1} \dots b_0)_2$.

Prova. ***Definições e Propriedades da ABC:***

- *Uma ABC com N nós tem altura $h = n = \lfloor \lg N \rfloor$.*
- *Todos os níveis $0, 1, \dots, n-1$ estão completamente preenchidos, contendo $2^n - 1$ nós.*
- *O nível n (último nível) contém os $k = N - (2^n - 1) = N - 2^n + 1$ nós restantes, preenchidos da esquerda para a direita.*
- *O número total de nós é $N = 1(raiz) + N_1 + N_2$.*
- *A representação binária de N é $N = \sum_{i=0}^{n} b_i 2^i$. Como $b_n = 1$, temos $N = 2^n + \sum_{i=0}^{n-1} b_i 2^i = 2^n + (b_{n-1} \dots b_0)_2$.*
- *O número de nós no último nível é $k = N - 2^n + 1 = (2^n + (b_{n-1} \dots b_0)_2) - 2^n + 1 = (b_{n-1} \dots b_0)_2 + 1$.*

Tamanhos das Subárvores N_1 e N_2: *A subárvore esquerda T_1 consiste nos nós da subárvore esquerda completa até o nível $n-1$, mais os nós que "caem"na subárvore esquerda no último nível.*

- *Nós em T_1 dos níveis 1 a $n-1$ (relativos a T): $\frac{(2^n-1)-1}{2} = 2^{n-1} - 1$.*
- *Nós em T_1 do nível n (último nível de T): Os primeiros $\min(k, 2^{n-1})$ nós do último nível k vão para a subárvore esquerda (pois a capacidade da subárvore esquerda no nível n é 2^{n-1}).*
- *Portanto, $N_1 = (2^{n-1} - 1) + \min(k, 2^{n-1})$.*

Date: 13 de Abril de 2025.

A subárvore direita T_2 consiste nos nós da subárvore direita completa até o nível $n-1$, mais os nós restantes do último nível.

- *Nós em T_2 dos níveis* 1 *a* $n-1$*:* $2^{n-1}-1$.
- *Nós em T_2 do nível n: Os nós restantes* $\max(0, k-2^{n-1})$.
- *Portanto,* $N_2 = (2^{n-1}-1) + \max(0, k-2^{n-1})$.

***Caso** $b_{n-1} = 1$**:** Se $b_{n-1} = 1$, então o valor $(b_{n-1} \dots b_0)_2$ é no mínimo 2^{n-1}.*

- $(b_{n-1} \dots b_0)_2 = 1 \cdot 2^{n-1} + (b_{n-2} \dots b_0)_2 \geq 2^{n-1}$.
- *Consequentemente,* $k = (b_{n-1} \dots b_0)_2 + 1 \geq 2^{n-1} + 1$*. Logo,* $k > 2^{n-1}$.

1 Agora, calculamos N_1 e N_2 sob esta condição:

- $N_1 = (2^{n-1}-1) + \min(k, 2^{n-1})$*. Como* $k > 2^{n-1}$, $\min(k, 2^{n-1}) = 2^{n-1}$.
  $N_1 = (2^{n-1}-1) + 2^{n-1} = 2 \cdot 2^{n-1} - 1 = \boxed{2^n - 1}$*. (Isto significa que a subárvore esquerda é uma árvore binária completa de altura $n-1$).*
- $N_2 = (2^{n-1}-1) + \max(0, k-2^{n-1})$*. Como* $k > 2^{n-1}$, $\max(0, k-2^{n-1}) = k - 2^{n-1}$.
  $N_2 = (2^{n-1}-1) + (k-2^{n-1}) = k-1$.
- *Substituindo* $k = (b_{n-1} \dots b_0)_2 + 1$*:*
  $N_2 = ((b_{n-1} \dots b_0)_2 + 1) - 1 = \boxed{(b_{n-1} \dots b_0)_2}$.

Assim, provamos que se $b_{n-1} = 1$, então $N_1 = 2^n - 1$ e $N_2 = (b_{n-1} \dots b_0)_2$.

2. Exercício 2

Usando a mesma notação do Exercício 1, queremos provar que se $b_{n-1} = 0$, então $N_1 = (1b_{n-2} \ldots b_0)_2$ e $N_2 = 2^{n-1} - 1$.

Prova. *Utilizamos as mesmas definições e as fórmulas gerais para N_1 e N_2 derivadas no Exercício 1:*

- $N = (1b_{n-1} \ldots b_0)_2$, $n = \lfloor \lg N \rfloor$.
- $k = N - 2^n + 1 = (b_{n-1} \ldots b_0)_2 + 1$.
- $N_1 = (2^{n-1} - 1) + \min(k, 2^{n-1})$.
- $N_2 = (2^{n-1} - 1) + \max(0, k - 2^{n-1})$.

Caso** $b_{n-1} = 0$**: *Se $b_{n-1} = 0$, então o valor $(b_{n-1} \ldots b_0)_2 = (0b_{n-2} \ldots b_0)_2 = (b_{n-2} \ldots b_0)_2$.*

- $(b_{n-1} \ldots b_0)_2 = \sum_{i=0}^{n-2} b_i 2^i < 2^{n-1}$.
- *Consequentemente,* $k = (b_{n-1} \ldots b_0)_2 + 1 \leq (2^{n-1} - 1) + 1 = 2^{n-1}$.

Agora, calculamos N_1 e N_2 sob esta condição:

- $N_1 = (2^{n-1} - 1) + \min(k, 2^{n-1})$. *Como* $k \leq 2^{n-1}$, $\min(k, 2^{n-1}) = k$.
  $N_1 = (2^{n-1} - 1) + k$.
- *Substituindo* $k = (b_{n-1} \ldots b_0)_2 + 1$*:*
  $N_1 = (2^{n-1} - 1) + ((b_{n-1} \ldots b_0)_2 + 1) = 2^{n-1} + (b_{n-1} \ldots b_0)_2$.
- *Como* $b_{n-1} = 0$, $(b_{n-1} \ldots b_0)_2 = (b_{n-2} \ldots b_0)_2$.
  $N_1 = 2^{n-1} + (b_{n-2} \ldots b_0)_2$.
- *Este valor é precisamente a representação binária* $\boxed{(1b_{n-2} \ldots b_0)_2}$. *(O 1 está na posição $n - 1$).*
- $N_2 = (2^{n-1} - 1) + \max(0, k - 2^{n-1})$. *Como* $k \leq 2^{n-1}$, $\max(0, k - 2^{n-1}) = 0$.
  $N_2 = (2^{n-1} - 1) + 0 = \boxed{2^{n-1} - 1}$. *(Isto significa que a subárvore direita é uma árvore binária completa de altura $n - 2$).*

Assim, provamos que se $b_{n-1} = 0$, então $N_1 = (1b_{n-2} \ldots b_0)_2$ e $N_2 = 2^{n-1} - 1$.

3. Exercício 3

Seja T uma ABC com N nós. Seja $h(x)$ a altura do nó x em T (definida como o comprimento do caminho mais longo de x até uma folha na subárvore enraizada em x). Seja $S_N = \sum_{x \in T} h(x)$ a soma das alturas de todos os nós em T. Seja u_N o número de bits 1 na representação binária de N. Queremos provar que $S_N = N - u_N$.

Prova. *Procederemos por indução forte sobre o número de nós N.*
Base da Indução: *Para $N = 1$.*

- *A árvore T consiste apenas no nó raiz r.*
- *A representação binária é $N = (1)_2$.*
- *O número de bits 1 é $u_1 = 1$.*
- *A altura da raiz (que também é folha) é $h(r) = 0$.*
- *A soma das alturas é $S_1 = h(r) = 0$.*
- *A fórmula $N - u_N$ resulta em $1 - u_1 = 1 - 1 = 0$.*
- *Como $S_1 = N - u_N$, a base da indução é válida.*

Hipótese Indutiva (HI): *Assuma que para todo k tal que $1 \leq k < N$, a soma das alturas S_k em uma ABC com k nós satisfaz $S_k = k - u_k$.*
Passo Indutivo: *Considere uma ABC T com $N > 1$ nós. Seja r a raiz de T. Sejam T_1 e T_2 as subárvores esquerda e direita com N_1 e N_2 nós, respectivamente, onde $N = 1 + N_1 + N_2$.*

- *A altura da raiz r é $h(r) = n = \lfloor \lg N \rfloor$.*
- *A soma das alturas em T pode ser calculada recursivamente: A altura de cada nó x na subárvore T_i (onde $i = 1$ ou $i = 2$) contribui para a soma S_N. A soma das alturas dos nós dentro de T_1 (calculadas como se T_1 fosse uma árvore independente) é S_{N_1}. Similarmente para T_2, a soma é S_{N_2}. A altura da raiz $h(r)$ deve ser adicionada.*
- *Assim, a relação recursiva é: $S_N = h(r) + S_{N_1} + S_{N_2}$.*
- *Note que $N_1 \geq 0$ e $N_2 \geq 0$. Como $N > 1$, pelo menos uma subárvore não é vazia. Se $N_1 = 0$ (ou $N_2 = 0$), então $S_{N_1} = 0$ (ou $S_{N_2} = 0$), e a HI ainda se aplica (formalmente, $u_0 = 0$, então $S_0 = 0 - 0 = 0$). Como $N \geq 1$, temos $N_1 < N$ e $N_2 < N$.*
- *Aplicando a Hipótese Indutiva para S_{N_1} e S_{N_2} (assumindo $N_1, N_2 \geq 1$, ou tratando $S_0 = 0, u_0 = 0$ se N_1 ou N_2 for 0):*

$$S_N = n + (N_1 - u_{N_1}) + (N_2 - u_{N_2})$$

- *Usando $N_1 + N_2 = N - 1$:*

$$S_N = n + (N - 1) - (u_{N_1} + u_{N_2})$$

Queremos mostrar que $S_N = N - u_N$. Para isso, precisamos verificar se a seguinte igualdade é verdadeira:

$$n + (N - 1) - (u_{N_1} + u_{N_2}) = N - u_N$$

Rearranjando os termos, a igualdade acima é equivalente a:

$$u_N = u_{N_1} + u_{N_2} - n + 1$$

Vamos verificar esta relação usando os resultados dos Exercícios 1 e 2. Seja $N = (1b_{n-1} \ldots b_0)_2$. O número de bits 1 em N é $u_N = 1 + \sum_{i=0}^{n-1} b_i$.
Caso 1: $b_{n-1} = 1$.

- *Do Exercício 1,* $N_1 = 2^n - 1 = (11\ldots1)_2$ *(n uns) e* $N_2 = (b_{n-1}\ldots b_0)_2 = (1b_{n-2}\ldots b_0)_2$.
- $u_{N_1} = n$.
- u_{N_2} *é o número de bits 1 em* $(b_{n-1}\ldots b_0)_2$*, que é* $\sum_{i=0}^{n-1} b_i$.
- *Verificando a relação:* $u_{N_1} + u_{N_2} - n + 1 = n + \left(\sum_{i=0}^{n-1} b_i\right) - n + 1 = 1 + \sum_{i=0}^{n-1} b_i$.
- *Este resultado é exatamente* $u_N = 1 + \sum_{i=0}^{n-1} b_i$. *A relação é válida neste caso.*

Caso 2: $b_{n-1} = 0$.

- *Do Exercício 2,* $N_1 = (1b_{n-2}\ldots b_0)_2$ *e* $N_2 = 2^{n-1} - 1 = (11\ldots1)_2$ *(n − 1 uns).*
- u_{N_1} *é o número de bits 1 em* $(1b_{n-2}\ldots b_0)_2$. *O bit mais significativo (posição* $n-1$*) é 1. Os outros bits são* $b_{n-2}, \ldots, b_0$. *Então* $u_{N_1} = 1 + \sum_{i=0}^{n-2} b_i$.
- $u_{N_2} = n - 1$.
- *Verificando a relação:* $u_{N_1} + u_{N_2} - n + 1 = \left(1 + \sum_{i=0}^{n-2} b_i\right) + (n-1) - n + 1 = 1 + \sum_{i=0}^{n-2} b_i + n - 1 - n + 1 = 1 + \sum_{i=0}^{n-2} b_i$.
- *Agora, vejamos* u_N. *Como* $N = (1b_{n-1}\ldots b_0)_2$ *e* $b_{n-1} = 0$: $u_N = 1 + b_{n-1} + \sum_{i=0}^{n-2} b_i = 1 + 0 + \sum_{i=0}^{n-2} b_i = 1 + \sum_{i=0}^{n-2} b_i$.
- *Novamente, o resultado* $u_{N_1} + u_{N_2} - n + 1$ *é igual a* u_N. *A relação é válida neste caso também.*

Como a relação $u_N = u_{N_1} + u_{N_2} - n + 1$ *foi verificada em ambos os casos possíveis para* b_{n-1}*, a equação* $S_N = n + (N-1) - (u_{N_1} + u_{N_2})$ *se simplifica corretamente para* $S_N = N - u_N$.

Conclusão da Indução: *Pelo princípio da indução forte, provamos que para qualquer ABC* T *com* $N \geq 1$ *nós, a soma das alturas de todos os nós é* $S_N = N - u_N$.