Análise de algoritmos - Trabalho Propriedades Heaps*

Eduardo Barbosa de Oliveira, Rafael Rampin Soratto
Coordenação do Curso de Bacharelado em Ciência da Computação - COCIC
Universidade Tecnológica Federal do Paraná - UTFPR
Campus Campo Mourão
Campo Mourão, Paraná, Brasil
eduardooliveira.1997@alunos.utfpr.edu.br e soratto@alunos.utfpr.edu.br

Resumo

O presente trabalho irá provar propriedades heap. Para isso, utilizaremos indução matemática e vários outros meios

1 Questões

Serão provadas as seguintes afirmações:

1. Prove que existem, no máximo, 2^h folhas em uma árvore binária de altura h.
2. Prove que uma árvore binária completa de altura h tem $2^h - 1$ nós internos.
3. Prove que uma árvore binária completa de altura h tem $2^{h+1} - 1$ nós.
4. Prove que uma árvore heap com n nós tem altura $\lfloor \lg n \rfloor$
5. Mostre que, com uma representação vetorial de uma heap com n elementos, o índice dos nós internos é no máximo $\lfloor \frac{n}{2} \rfloor$.
6. Mostre que, com uma representação vetorial de uma heap com n elementos, as folhas são nós indexados por $\lfloor \frac{n}{2} + 1 \rfloor$, $\lfloor \frac{n}{2} + 2 \rfloor$, ..., n
7. Prove que um vetor ordenado é um heap mínimo.

2 Provas

2.1 Questão 1

Para provar que existem, no máximo, 2^h folhas em uma árvore binária, iremos utilizar indução matemática. Para isso, seja l a quantidade de nós folhas em uma árvore binária, temos que nossa hipótese é:

$$l \leq 2^h \text{ (Hipótese)}$$

Caso base: vamos fazer para h = 0. Uma árvore binária com altura 0 é uma árvore com apenas um nó, ou seja, apenas um nó folha. Então, temos que:

$$\begin{aligned} &2^h = 2^0 = 1 \\ &l = 1 \\ &logo, l \leq 2^h \end{aligned} \tag{1}$$

Logo, como mantivemos a hipótese indutiva para um valor de h, temos que o caso base foi concluído e verificado.

Hipótese indutiva: suponhamos que para uma altura k qualquer, todas as árvores binárias da altura $\leq$ k, tenham no máximo 2^k nós folhas. Para k+1 temos que:

$$l = 2^{k+1}$$

Sendo A uma árvore binária de altura k+1. Então suas subárvores da esquerda e da direita tem cada uma no máximo 2^k nós folhas, pela hipótese indutiva. Logo, a quantidade l de nós folhas de A é dada pela soma de nós folhas de suas subárvores. Sendo assim, temos que:

$$\begin{aligned} &l \leq 2^k + 2^k [H.I] \\ &l \leq 2 * 2^k \\ &l \leq 2^{k+1} \end{aligned} \tag{2}$$

Então, como l ainda é menor que 2^{k+1}, e k+1 é a altura da árvore, nossa hipótese indutiva se manteve para k+1, sendo assim, o teorema foi provado pelo método da indução matemática.

2.2 Questão 2

Iremos provar por indução matemática que uma árvore binária completa de altura h tem $2^h - 1$ nós internos. Então seja n o número de nós internos da árvore binária, temos que:

$$n = 2^h - 1 \text{ (Hipótese)}$$

Caso base: faremos para h = 1. Para isso, temos que nossa árvore binária completa tem 3 nós. Sendo um deles o raíz e dois deles nós folhas. Sendo assim, obtivemos que:

$$n = 2^h - 1 = 2^1 - 1 = 2 - 1 = 1$$

Sendo assim, como nossa árvore só tem um nó interno (o raiz, no caso), e obtivemos pela nossa equação o valor 1, temos que o caso base foi verificado e concluído.

Hipótese indutiva: suponhamos que nossa propriedade se mantém para qualquer k ≥ 0. Para k+1 temos então que:

$$n = 2^{k+1} - 1$$

Sendo T uma árvore binária complete de altura k+1. Se removermos todos os nós folhas, temos uma árvore binária completa de altura k. Sabemos que uma árvore binária completa tem no máximo 2^h nós folhas no máximo. Como se trata de uma árvore binária completa, sabemos que, no nível k, temos exatamente 2^k nós fohas. Sendo assim, a quantidade de nós total do nível k (que é a quantidade total de nós internos da árvore binária completa com altura k+1) é dada pela soma da quantidade de nós folhas e quantidade de nós internos da árvore binária completa de altura k:

$$\begin{aligned} &n = 2^k + 2^k - 1[H.I] \\ &n = 2 * 2^k - 1 \\ &n = 2^{k+1} - 1 \end{aligned} \tag{3}$$

Sendo assim, conseguimos provar que a propriedade se mantém para k+1, logo, provamos por indução matemática que o número de nós internos de uma árvore binária completa de altura h é $2^h - 1$.

2.3 Questão 3

Iremos utilizar indução matemática para provar que uma árvore binária completa de altura h tem $2^{h+1} - 1$ nós. Seja n a quantidade de nós, então nossa hipótese indutiva é:

$$n = 2^{h+1} - 1 \text{ (Hipótese)}$$

Caso base: faremos para h = 0. Para termos isso, precisamos ter apenas um único nó, que será o raiz. Sendo assim:

$$\begin{aligned} &2^{h+1} - 1 = 2^{0+1} - 1 = 2^1 - 1 = 1 \\ &n = 1 \\ &logo, n = 2^{h+1} - 1 \end{aligned} \tag{4}$$

Como n = $2^{h+1} - 1$, temos que o caso base foi concluído e verificado.

Hipótese indutiva: suponha que a propriedade se mantém para qualquer k $\geq$ 0.Para k+1 temos então que:

$$n = 2^{(k+1)+1} - 1 = 2^{k+2} - 1$$

Então, seja T uma árvore binária de altura k+1, por definição, todos os nós no nível k+1 são folhas. Pela Questão 1, vista na subseção 2.1, sabemos que uma árvore binária tem no máximo 2^{k+1} nós folhas. Como a árvore é completa, sabemos que a quantidade de nós folhas é exatamente 2^{k+1}. Se removermos todos os nós folhas do nível k+1, nos resta uma árvore com altura k. Então, somando o número de nós folhas do nível k+1 e o número de nós da árvore com altura k, temos que:

$$\begin{aligned} &n = 2^{k+1} + 2^{k+1} - 1[H.I.] \\ &n = 2 * 2^{k+1} - 1 \\ &n = 2^{k+2} - 1 \end{aligned} \tag{5}$$

Sendo assim, chegamos que a propriedade se mantém para k+1, então, provamos que uma árvore completa tem $2^{k+1} - 1$ nós no total, por indução matemática.

2.4 Questão 4

Para provar que uma árvore *"heap"* com "**n**"nós tem altura $\lfloor$**lg n**$\rfloor$ iremos associar um **heap** com uma árvore binária completa com exceção do último nível.

$$h = \log n \text{ (Hipótese)}$$

Podemos então criar a representação de níveis com números de nós por níveis para saber a relação entre ambos.

Nível	Número de Nós
0	1
1	2
2	4
3	8
h	2^h

Tabela 1: Tabela (nível / quantidade de nós)

De acordo com a tabela 1 podemos chegar na equação:

$$\begin{aligned} h &= 2^h \\ \log h &= \log 2^h \\ \log h &= h \end{aligned} \tag{6}$$

Theorem 2.1 *Em uma arvore binária de altura h existem no máximo* $(2^{h+1} - 1)$ *elementos, e no mínimo* 2^h*.*

$$\begin{aligned} 2^h \leq n & \qquad \leq (2^{h+1} - 1) \\ \log(2^h) \leq \log n & \leq \log(2^{h+1} - 1) \\ h \leq lg(n) & \qquad \leq h + 1 \end{aligned}$$

Para provar o teorema 2.1 iremos considerar uma altura h = 2. Então teremos a seguinte árvore resultante:

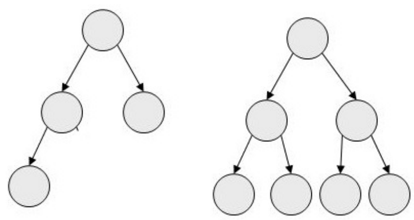

Figura 1: Quantidade Mínima e máxima para h = 2

Possui no máximo 2^{h+1} nós $((2^{2+1}) - 1 = 7)$.
E no mínimo 2^h nós $((2^2) = 4)$. Concluindo que para todo valor n, e uma altura h exitem no máximo 2^{h+1} e no mínimo 2^h nós em sua "Arvore binária".

2.5 Questão 5

Definição 2.1

Elemento de um Heap : Todo elemento em um vetor heap de tamanho n que estiver em uma posição até $[n/2]$ *é considerado nó intermediário, portanto possui um ou mais filhos. (e.g.(*$[(n/2) - 1]$*).*

São eles:

- $(2 * i)$ Se e somente se :

$$2 * i \leq n$$

- $(2 * i) + 1$. Se e somente se :

$$(2 * i) + 1 \leq n$$

- Nós folhas(não tem filhos). Ou seja:

$$n < (2 * i)$$

  E isso garante que do intervalo de $[n/2, n/2+1, n/2+2, ..., n]$ todos os elementos sejam folhas da árvore.

2.6 Questão 6

Para mostrar que com uma representação vetorial de uma heap com n elementos, o índice dos nós internos é no máximo $\lfloor \frac{n}{2} \rfloor$. De acordo com 2.5 os nós intermeários possuem filhos com duas vezes seu indice ou duas vezes seu indice mais uma unidade. Ainda em 2.5 foi mostrado que "Nós folhas"não tem filhos. Ou seja, não existe posição no vetor do heap para qualquer posição a frente da metade do vetor tenha algum filho:

$$n < (2 * i)$$

2.7 Questão 7