

Projeto de Algoritmo com Implementação nº 0

MC458 - 2s2020 - Tiago de Paula Alves - 187679

1 Definições

Um móbile M , com objetos E e D presos na barra, é dito em equilíbrio se $P_e \cdot D_e = P_d \cdot D_d$ e todos os seus submóviles também estão em equilíbrio, sendo que cada objeto E ou D pode ser um submóBILE ou um peso simples. No entanto, essa definição só faz sentido para móveis, mas ela pode ser expandida para pesos simples, considerando esse último tipo de objeto como sempre em equilíbrio.

Com isso, podemos definir equilíbrio de forma recursiva, mas equivalente, considerando os dois tipos de objetos (móviles ou pesos simples):

Equilíbrio: Dizemos que um objeto O está em equilíbrio se uma destas condições for atendida:

- i) O é um pesos simples.
- ii) O é um móBILE com os objetos E e D presos, tal que E e D estão em equilíbrio e $P_e \cdot D_e = P_d \cdot D_d$.

Essa definição requer que o peso total P_O de um objeto O seja conhecido, o que é verdade apenas para pesos simples. Para suprir todos os casos podemos definir uma função $P : \text{Objeto} \rightarrow \mathbb{N}$ que calcula o peso total de um objeto.

$$P(O) = \begin{cases} P_O & \text{se } O \text{ é um peso simples} \\ P(E) + P(D) & \text{se } O \text{ é um móBILE com objetos } E \text{ e } D \end{cases}$$

2 Implementação

Os tipos usados na implementação podem ser definido de forma mais abstrata como:

```
struct Mobile {  
    E, D    :: Objeto  
    De, Dd  :: Natural  
}
```

```
tipo Peso = Natural  
  
tipo Objeto = Peso | Mobile
```

Podemos ver que um móBILE M funciona como uma árvore binária especializada, onde cada submóBILE é um nó interno (com exatamente dois filhos) e cada pesos simples é uma folha. Então, a entrada foi lida em uma árvore de objetos, sendo que os pesos simples foram tratados apenas como um inteiro, o seu peso fixo P_O .

3 Algoritmo

O algoritmo para descobrir se um móBILE está em equilíbrio pode ser deduzido a partir de uma indução forte, como é comum para árvores binárias. No entanto, precisamos também dos pesos dos os objetos presos no móBILE.

Então, podemos fortalecer a hipótese com o cálculo da função P . Além disso, podemos usar a definição recursiva de equilíbrio, expandindo a hipótese para objetos em geral.

Hipótese Fortalecida. Dado um objeto O com n massas simples, podemos calcular seu peso total $P(O)$ e verificar se ele está em equilíbrio.

Note que a indução é em n , o número de massas no objeto. Como apenas pesos simples têm massa considerável, n também é o número de pesos simples compreendido no objeto, ou também o número de folhas na árvore.

Demonstração. Suponha um inteiro $n \geq 1$ tal que para todo objeto S composto de $1 \leq k < n$ massas podemos descobrir seu peso total $P(S)$ e se S está em equilíbrio. Suponha ainda um objeto O com n massas.

Caso 1: O é um peso simples, ou seja, $n = 1$. Logo, o peso dele é conhecido, então seja P_O esse peso. Portanto, $P(O) = P_O$ e, por definição, O está em equilíbrio.

Caso 2: O é um móbile, então, $n \geq 2$. Seja E o objeto peso na ponta esquerda de O , à uma distância D_e do sustentáculo, e D o objeto na direita, com distância D_d . Além disso, E terá n_e massas e D , n_d , sendo que $n_e + n_d = n$, já que a barra e a sustentação do móbile não têm peso.

Note que D deve ter $n_d \geq 1$ massas, portanto, E terá $1 \leq n_e = n - n_d \leq n - 1 < n$ massas. Logo, pela hipótese indutiva, podemos calcular $P(E)$ e sabemos se E está em equilíbrio. Da mesma forma, $1 \leq n_d < n$. Então, podemos conseguir $P(D)$ e saber se D está em equilíbrio.

Assim, se E e D estão em equilíbrio e $P(E) \cdot D_e = P(D) \cdot D_d$, então O também está em equilíbrio. Caso alguma dessas condições não seja atendida, O não está em equilíbrio. Portanto, teremos que $P(O) = P(E) + P(D)$ e podemos dizer se O está em equilíbrio. ■

O algoritmo derivado da demonstração acima é um algoritmo de Divisão e Conquista, que pode ser explicitado nas seguintes etapas:

Divisão: Separa um móbile em seus objetos, presos nas extremidades da barra.

Conquista: Descobre recursivamente o peso dos subobjetos e se eles estão em equilíbrio.

Combinação: O peso total do móbile é a soma dos pesos dos objetos e ele estará em equilíbrio se os objetos estão em equilíbrio e se a equação de equilíbrio for garantida.

4 Complexidade

Podemos ver pelo algoritmo que a etapa de divisão, em uma estrutura de árvore, segue trivialmente, com tempo constante. A etapa de combinação também é constante, fazendo apenas operações lógicas e aritmética independentes da entrada. Podemos chamar de $t_2 > 0$ o tempo dessas duas etapas. A única etapa com tempo dependente da entrada, neste algoritmo, é a conquista, implementada recursivamente. Além disso, o teste para um peso simples tem tempo constante $t_1 > 0$.

Assim, teremos que tempo de execução para um objeto O será:

$$T(O) = \begin{cases} t_1 & \text{se } O \text{ for um peso simples} \\ T(E) + T(D) + t_2 & \text{se } O \text{ for um móbile, sendo } E \text{ e } D \text{ seus objetos} \end{cases}$$

Agora, considerando o tamanho da entrada n como a quantidade de massas distribuídas no objeto, teremos que um peso simples é apenas uma massa, ou seja, $n = 1$. Além disso, um móbile deve ter dois objetos E e D presos nele, compreendendo $n_e \geq 1$ e $n_d \geq 1$ massas, respectivamente, de modo que $n_e + n_d = n \geq 2$. Então:

$$T(n) = \begin{cases} t_1 & \text{se } n = 1 \\ T(n_e) + T(n_d) + t_2 & \text{se } n \geq 2 \end{cases}$$

Podemos ver que o cálculo dos pesos depende de todas as n massas, então vale supor que o algoritmo é da classe $\Theta(n)$. Vamos provar então a fórmula fechada $T(n) = cn + b$ por indução forte em n .

Demonstração. Considere as constantes $c = t_1 + t_2$ e $b = -t_2$.

Caso base: Seja $n = 1$. Logo, $T(1) = c + b = t_1 + t_2 - t_2 = t_1$, como esperado.

Hipótese indutiva: Suponha que $T(k) = ck + b$ para todo $1 \leq k < n$ e algum $n \geq 2$.

Passo indutivo: Seja $n \geq 2$. Logo, o objeto deve ser um móbile, com filhos E e D , compostos de n_e e n_d massas simples. Como o móbile não têm massa própria, então $n_e + n_d = n$. Além disso, $1 \leq n_e < n$ e $1 \leq n_d < n$, então podemos aplicar a hipótese indutiva:

$$\begin{aligned} T(n) &= T(n_e) + T(n_d) + t_2 \\ &= cn_e + b + cn_d + b + t_2 \\ &= c(n_e + n_d) + b + (b + t_2) \\ &= cn + b \end{aligned}$$

A partir da demonstração acima e como $c > 0$, podemos concluir que $T(n) \in \Theta(cn + b) = \Theta(n)$. Se considerarmos o número de móveis m no sistema, teremos que $m = n - 1$, que matém a complexidade. ■