



Organizando um Baile: Solução do Problema da Organização de uma festa

Erick Grilo
Max Fratane
Matheus Prado
Vitor Santos

1 Introdução

O problema consiste em que o professor Stewart foi contratado para prestar um serviço de consultoria para o presidente de uma determinada empresa que deseja realizar uma festa e nós vamos ajudá-lo. A empresa possui uma hierarquia tal que a relação de hierarquia forma uma árvore cuja raiz é o presidente da empresa. O departamento do RH classificou cada um dos empregados com uma classificação de conviviabilidade (que é um número real). Para a festa ser proveitosa o máximo para todos que forem, o presidente não quer que ambos um funcionário e seu supervisor vão simultaneamente.

Ao professor Stewart, é dada uma árvore onde cada nó da árvore é um funcionário da companhia, que possui um nome e um valor de conviviabilidade, onde o pai desse nó é o seu supervisor imediato. O objetivo é criar um algoritmo que cria uma lista de convidados que maximiza a soma dos valores de conviviabilidade dos convidados.

2 O Algoritmo

A ideia do algoritmo consiste em solucionar o problema para a árvore de funcionários, com dois casos: caso a árvore tenha somente um nó, e caso a árvore tenha mais de um nó. Dessa forma, a partir do enunciado, temos:

A partir da descrição do trabalho acima, é possível deduzir a seguinte relação de recorrência:

$$Baile(tree) = \begin{cases} \text{funcionário}(tree), \text{ caso tree não tenha filhos} \\ \text{MAX} = \begin{cases} \sum Baile(t), \forall t \in \text{filho}(tree) \\ \text{conviviabilidade}(raiz) + \sum Baile(t), \forall t \in \text{filho}(\text{filho}(tree)) \end{cases} \end{cases} \quad (1)$$

Onde o caso base é o caso de *tree* ser uma árvore com um só nó e a relação de recorrência consiste no caso de *tree* tenha filhos: temos que avaliar então, dentre todos os filhos da árvore de entrada, quais serão os nós da árvore que maximizarão o somatório de conviviabilidade, tomando a precaução de não permitir que um empregado e seu supervisor imediato possam ir ao baile. Nesse caso, na primeira linha da parte do MAX, é avaliado o caso dos filhos de *tree* (caso o nó da árvore atual vá à festa), e na segunda linha, a conviviabilidade do nó imediatamente acima de *tree* e os filhos dos seus filhos, ou seja, o funcionário do funcionário da empresa. Dessa forma, é evitado que um funcionário e seu supervisor imediato possam ir ao baile. A função *funcionário* retorna o funcionário de um determinado nó, enquanto a função *conviviabilidade* retorna o valor de conviviabilidade de um determinado nó.

Da relação de recorrência apresentada acima, temos o seguinte pseudo-código, que ilustra como o algoritmo funciona:

Entrada: tree: uma árvore com os funcionários da empresa, Lista: lista de convidados, inicialmente vazia

Saída: Lista com os convidados com a maior soma de valores de convivabilidade possível, respeitando a restrição, e a soma dos valores de convivabilidade dos elementos da lista.

Baile(tree,lista):

```

se tree não possui filhos então
    | lista ← adicionar funcionario(tree)
    | retorna lista,conviviabilidade(tree)
senão
    | A ← 0.0
    | B ← conviviabilidade(tree)
    | para cada filho x ∈ filho(tree) faça
    | | A ← A + Baile(x)
    | | para cada filho y ∈ filho(x) faça
    | | | B ← B + Baile(y)
    | | fim
    | fim
    | se B > A então
    | | lista ← lista + funcionario(tree)
    | | retorna lista,B
    | fim
fim

```

Algoritmo 1: BAILE