

## Relatório Problema de Programação 2

### Equipa:

Nº Estudante: 2018283864 Nome David Marcelino Mendes Palaio

Nº Estudante: 2018283200 Nome Rui Alexandre Coelho Tapadinhas

### 1. Descrição do Algoritmo

Para resolver este problema considerámos uma abordagem recursiva bottom-up, em que cada chamada da função compara qual o melhor resultado: usar todos os filhos desse nó ou usar apenas esse nó, depois, guardamos no array *ratings*, o número de nós selecionados e o custo de cada um dos processos, a seguir sobe-se na árvore, e os nós pais, realizam a mesma operação usando os valores de *ratings* já atualizados dos nós filhos.

### 2. Estruturas de Dados

O nosso algoritmo consiste na manipulação de dois objetos do tipo `std::map<K, V>` da linguagem C++:

Um dos mapas tem o nome de *pyramid\_scheme* que armazena todos os pares recrutador/recrutado, sendo os K os recrutadores e os V um vetor de recrutados. O último valor presente no V de cada K representa o valor que essa pessoa pagou para entrar na organização.

O segundo mapa chama-se *ratings* e armazena para cada nó (chave) um vetor de vetores que contém um tamanho fixo, corresponderá a uma matriz 2x2. A primeira linha destina-se a guardar o número de nós visitados e o dinheiro pago por esses nós quando a chave correspondente não é usada, enquanto a segunda guardará os mesmos valores mas para a situação em que o nó é usado.

### 3. Correção

Pensamos que obtivemos 200 pontos no Mooshak pelas seguintes razões:

- Respondemos corretamente ao problema em questão;
- Apresentámos uma solução eficiente pois conseguimos construir um algoritmo com uma complexidade temporal quadrática, algo que dificilmente conseguiríamos melhorar pois não encontramos forma de resolver o problema sem passar obrigatoriamente por todos os nós recrutados e fazer uma chamada recursiva para o mesmo número nós.
- Conseguimos simplificar o nosso algoritmo ao não termos de saber quais são os nós a serem usados, guardando apenas os resultados da contagem de nós e o custo dessa solução.

### 4. Análise Algoritmo

Complexidade Temporal:  $O(n^2)$ , sendo  $n$  o número de recrutados.

Complexidade Espacial:  $O(s+n)$ , sendo  $s$  o número de recrutados por recrutador e  $n$  o número de recrutadores.