

Trabalho Prático 2: The Force Awakens

Gabriel Henrique Souto Pires {gabrielpires@ufmg.br}

1 Introdução

Após 30 anos de paz na galáxia, o lado negro da força planeja seu retorno e você como o novo Lord Sith planeja reconquistar a galáxia, vingar a morte de seu mestre Darth Vader e terminar o que ele começou. O problema proposto consiste em conquistar um número k de planetas usando a sua nova super arma galáctica, a Estrela da Morte III. Com um mapa da galáxia em mãos é possível criar uma rota que começa no ponto atual da Estrela da Morte III, passa por todos os planetas que podem ser conquistados nas proximidades contando com o valor da distância entre eles e terminando em um ponto de chegada. Como a nova estação do império ainda esta em fase de testes, é necessário conquistar os planetas usando o minimo de recursos possível, ou seja, se deslocando o mínimo possível entre cada planeta. Usando a rota traçada entre os planetas a serem conquistados, k planetas devem ser escolhidos para se conquistar de modo que a sub-distância mínima do percurso seja minimizada.

2 Solução do Problema

O problema proposto foi resolvido utilizando três paradigmas de programação, sendo eles força bruta, programação dinâmica e um algoritmo guloso.

2.1 Força Bruta

2.2 Programação Dinâmica

2.3 Algoritmo Guloso

Na solução gulosa foi utilizado um vetor para guardar as distâncias a serem percorridas. Nesse vetor cada posição representa um estágio do caminho a ser percorrido pela nave, sendo que a posição 0 é referente ao início do caminho e a posição $n + 2$ representa o final do caminho.

3 Análise Teórica do Custo Assintótico

A complexidade assintótica do programa será discutida na subseção abaixo.

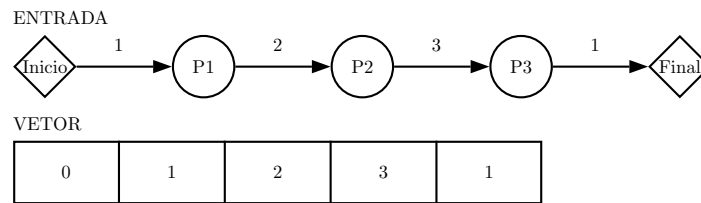


Figure 1: Formatação da entrada

3.1 Análise Teórica do Custo Assintótico de Tempo

3.1.1 Força Bruta

3.1.2 Programação Dinâmica

3.1.3 Algoritmo Guloso

3.2 Análise Teórica do Custo Assintótico de Espaço

3.2.1 Força Bruta

3.2.2 Programação Dinâmica

3.2.3 Algoritmo Guloso

4 Análise Experimental do Custo Assintótico

Para fazer a análise experimental do programa, foi criado um outro programa que gera arquivos de entrada compatíveis de tamanhos variados de forma que a diferença do tempo de execução com entradas diferentes possa ser medido. Para não levar em consideração o tempo que se leva para digitar a entrada no tempo de execução do programa, a entrada foi lida diretamente dos arquivos. Para se medir o tempo de execução do programa foi usado o comando *time* que ao final da execução do programa grava na saída padrão (*stdout*) estatísticas de tempo sobre o programa executado.

Os testes foram realizados em uma máquina virtual rodando Xubuntu 15.10. O computador utilizado tem um processador AMD Phenom II X4 965 3.40GHz e 8GB de memória, porém, a máquina virtual utiliza apenas um núcleo do processador e 2GB de RAM.

5 Conclusão

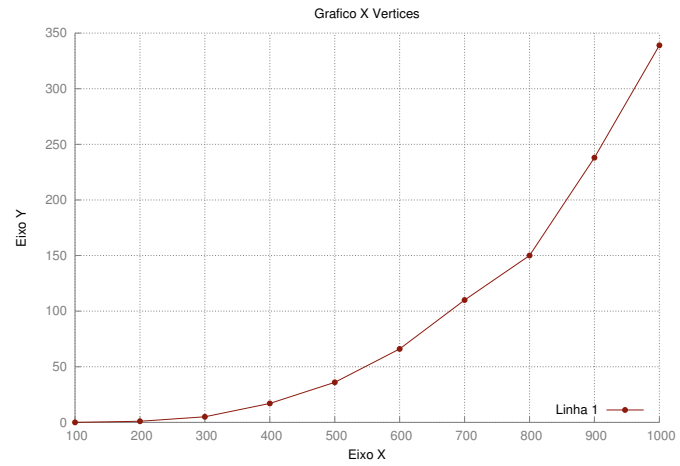


Figure 2: Gráfico da análise experimental