

Trabalho Prático III

As viagens de Johnny

Eduardo Capanema

Matrícula 2020041515

Aluno de Graduação

Matemática Computacional

Universidade Federal de Minas Gerais

Março, 2021

Submetido para Avaliação: Disciplina de Algoritmos I (Semestre 2020/2).

Documentação

O trabalho prático III da disciplina de Algoritmos I, do segundo semestre letivo do ano de 2020, nos solicitou que implementássemos, utilizando a linguagem de programação `c` ou `c++`, uma solução para um problema de ordem prática. Resumidamente, o problema consiste em determinar, a partir de uma entrada padronizada, o menor custo de viagens programadas usando uma escala para o metrô de Nova York.

1.1 | Modelagem Computacional

A modelagem computacional que julgamos adequada para a solução do problema consistiu em utilizar Programação Dinâmica (quando podemos subdividir um problema e utilizar persistência) de forma a reutilizar dados, através de recorrência e tornar a implementação mais prática do ponto de vista da complexidade temporal.

A partir do padrão de entrada dos dados, tratamos os dados para realizar um procedimento recursivo responsável por nos fornecer, diante dos inúmeros cenários de custo de viagens, a partir das escalas fornecidas, que atendem aos critérios e minimizam este custo.

1.2 | Estruturas de Dados

Em nossa implementação, utilizamos algumas estruturas de dados bastante comuns. A leitura dos dados foi feita utilizando `cin` e carregamento de seus valores em vetores, `vector<int>`.

1.2.1 | Programação Dinâmica

Nossa implementação, como dito anteriormente, se beneficiou do uso da técnica de Programação Dinâmica, que nos permitiu diminuir a complexidade temporal da programação. De acordo com a Wikipedia:

"Programação dinâmica é um método para a construção de algoritmos para a resolução de problemas computacionais, em especial os de otimização combinatória. Ela é aplicável a problemas nos quais a solução ótima pode ser computada a partir da solução ótima previamente calculada e memorizada - de forma a evitar recálculo - de outros subproblemas que, sobrepostos, compõem o problema original. O que um problema de otimização deve ter para que a programação dinâmica seja aplicável são duas principais características: subestrutura ótima e superposição de subproblemas. Um problema apresenta uma subestrutura ótima quando uma solução ótima para o problema contém em seu interior soluções ótimas para subproblemas. A superposição de subproblemas acontece quando um algoritmo recursivo re-examina o mesmo problema muitas vezes."

1.3 | Algoritmos

Embora não tenhamos utilizado nenhum algoritmo clássico, como Dijkstra ou DFS em Trabalhos Práticos realizados para a disciplina anteriormente, utilizamos um algoritmo que criamos, a partir do problema dado, e aplicamos Programação Dinâmica com recursividade para obter uma solução ótima e reduzida com relação à implementação do mesmo algoritmo sem persistir dados, conforme o paradigma em questão determina.

1.4 | Complexidade

Ao final de nossa implementação, realizamos a análise da complexidade assintótica, tanto do tempo quanto do espaço (uma vez que, conforme já mencionado, a recursividade foi utilizada, requerindo assim uma análise de espaço).

Nossa implementação mostrou-se prática uma vez que sua execução, em tempo e espaço, é polinomial, i.e. $O(n^2)$, visto que percorremos em dois loops do tipo *for* os dados recebidos.

1.5 | Conclusão

Nosso trabalho foi capaz de solucionar, para as entradas padrão fornecidas e outras que fomos capazes de testar, o problema proposto para a minimização dos custos de deslocamento de João da Silva em sua viagem para Nova York. Nosso algoritmo, que desenvolvemos utilizando Programação Dinâmica, permitiu que, por meio de recursividade, encontrássemos os melhores valores para os percursos escolhidos.