Trabalho Prático 3
Algoritmos 1 - 2022/2

Luiza de Melo Gomes - Turma TW

Matrícula: 2021040075
luizademelo@dcc.ufmg.br

1 Introdução

O trabalho implementado consiste na implementação da resolução do problema de que, dada uma remessa de rolos, em que se conhece a ordem e o preço dos produtos, qual é a maior quantidade de rolos que é possível expor de forma que respeite a ordem de inserção e ao mesmo tempo os rolos estejam na ordem decrescente de preço.

2 Modelagem

Para modelar o problema, houve a leitura dos rolos e a inserção de cada um tanto no início, quanto no final de um vetor `v[]` que armazena os rolos. Dessa forma, cada rolo será inserido duas vezes. Porém, isso facilita a resolução do problema, pois para testamos todas as possibilidades de inserção de um rolo (no início ou no fim da prateleira). Após isso, executamos um algoritmo que calcula qual é o tamanho da maior subsequência decrescente desse vetor.

2.1 Método de resolução

Em primeiro lugar, definimos um vetor `lds[0...r-1]` que armazena o tamanho da maior subsequência de rolos que termina no i-ésimo rolo. Esses tamanhos são calculados gradualmente: primeiro calculamos `lds[0]`, `lds[1]` ... até o rolo de index n-1. A solução para calcular o i-ésimo rolo envolve o conhecimento da solução do (i-1)-ésimo rolo. Sendo assim, armazenamos essas soluções em um vetor para evitar o cálculo desses subproblemas várias vezes.

2.2 Relação de Recorrência

Caso base: `lds[0]` = 1. Pode-se colocar um único rolo na prateleira.
Caso geral: `lds[n]` = max(`lds[i] + 1`, 1) para i = 0 até i = n-1 e para todo rolo `v[i]` tal que o seu preço seja maior do que o preço do n-ésimo rolo.

2.3 Análise de Complexidade

A código possui dois laços aninhados para realizar o cálculo da maior quantidade de rolos que conseguimos armazenar: para cada rolo, precisamos olhar para os outros que foram inseridos antes dele para checar se é possível fazer uma subsequência válida. Sendo assim, a complexidade do algoritmo é $O(n^2)$.

2.4 Funções e Variáveis

O programa possui somente duas funções: `main` e `lds`:
`lds`: recebe um vetor `v` contendo os rolos. Retorna a quantidade máxima de rolos que é possível expor

na prateleira. Para realizar esse cálculo, a função cria um vetor `lds[]` para armazenar as soluções dos i-ésimos elementos.
`main`: a função principal do programa realiza iterativamente a leitura das entradas do programa. Segue uma breve descrição das variáveis utilizadas:
`n`: quantidade de casos de teste.
`r`: quantidade de rolos utilizados em cada teste.
`p`: para cada rolo lido, o seu preço é armazenado em `p` e adicionado no vetor `v` de rolos.

3 Compilação e execução

Para compilar o programa, é necessário executar o comando `make`. Já para realizar a sua execucao com um determinado arquivo de texto `input.txt`, por exemplo, é necessário executar o comando `./tp03 < input.txt`.