Gabriel Chaves Ferreira
2018046700

Trabalho Pratico 3

Modelagem computacional do problema

O problema tem como objetivo solucionar um problema enfrentado por João, que deseja viajar com pouco dinheiro e para isso decidiu usar o cartão Big Apple que o ajuda a conseguir descontos no metrô de Nova York.

Para a sua viagem, existem diversas escalas no metrô nas quais ele tem que passar para chegar ao seu destino final. Com o cartao Big Apple, João recebe descontos em viagens cumulativas, desde que respeitado um limite de tempo e um limite de escalas subsequentes com descontos.

Temos como entrada **N** escalas nas quais João deve passar. Cada uma dessa escala também possui um tempo, que indica quanto tempo é gasto entre a escala **i** até a escala **i + 1**. Também recebemos como entrada um limite de tempo, **LT**, para o desconto cumulativo e o limite de escalas subsequentes, **LS**.

Baseado nisso podemos construir uma matriz para representar o problema, onde temos **i** linhas que representam as **N** escalas, e **j** colunas que representam o desconto aplicado. Por exemplo, se temos como entrada 10 escalas, e um limite **LS** de 6, nossa matriz será uma matriz 10 x 6. Essa matriz é construída baseado no conceito de programação dinâmica, onde a posição **ij** é obtida respondendo os seguintes subproblemas:

1. Se **j** for igual a 0, o valor de **ij** será o preço da escala **i** aplicado o primeiro valor dos descontos cumulativos mais o menor valor entre todas as colunas da linha **i - 1.**
2. Se **j** for diferente de 0, o valor de **ij** será o preço da posição **i - 1** e **j - 1** mais o valor da escala **i** aplicado o desconto cumulativo **j** desde que não se tenha ultrapassado o limite de tempo. Caso o limite de tempo tenha estourado, o valor de **ij** será o preço da posição **i - 1** e **j - 1** mais o valor da escala **i** com o primeiro desconto aplicado.

Isso nos leva a concluir que o menor preço para a viagem será o menor preço que encontrarmos na linha mais alta da nossa matriz. Por exemplo, se nossa matriz tem 10 linhas, o menor preço presente nas colunas da linha 10 será o nosso resultado.

Estrutura de dados e algoritmos utilizados:

Meu algoritmo tem três estruturas de dados (podem ser encontradas no arquivo **estruturas.h**:

1. Escala: Uma classe para representar as escalas, com seus respectivos preços e tempo gasto no trajeto da escala.
2. Desconto: Uma classe para representar uma posição **ij** da matriz e facilitar a modelagem do problema, nela armazeno o preço e tempo acumulado até a escala **i** com desconto acumulado **j**.

O algoritmo utilizado para resolver o problema segue da seguinte forma:

1. Insiro na primeira posição da matriz (conside como 1 1 para ficar igual ao número das escalas) 1 1 da matriz o valor e tempo da escala 1 com o primeiro desconto aplicado. Note que aqui i = 1 e j = 1.

2. Calculo os valores para a escala i + 1 e todas as suas colunas j + 1, j + 2 ... ate J = **limite de descontos cumulativos**, utilizando os subproblemas descritos anteriormente na modelagem computacional do problema.

3. Repito o processo em 2 até ter preenchido a última coluna da matriz, que equivale a linha **N**.

Análise de complexidade assintótica do problema:

1. Para o preenchimento da matriz, o algoritmo executa um for **N** vezes, onde n é a quantidade de escalas. E dentro dele temos um for que executa **M** vezes, onde m é o limite de descontos cumulativos.
2. No final temos um algoritmo que roda em complexidade **O(n*m)**, onde **n** é a quantidade de escalas e **m** o limite de descontos cumulativos.

Como compilar e executar:

O compilador é o g++.

1. make clean
2. make
3. ./tp03