Universidade Federal de Viçosa - Campus Florestal

Lucas Barros 3511
Vinícius Júlio 3495

Algoritmos e Estruturas de Dados I

Análise de Complexidade do Problema da Mochila

Florestal - MG
6 de Novembro 2019

Lucas Barros 3511
Vinícius Júlio 3495

Análise de Complexidade do Problema da Mochila

2° trabalho prático de Algoritmos e Estruturas de Dados 1. Trabalho que visa a concretizar conceitos de Análise da Complexidade de Algoritmos através da aplicação do Problema da Mochila.

Professora: Prof ª. Thais R. M. Braga Silva
Disciplina: Algoritmos e Estruturas de Dados 1

Florestal - MG
6 de Novembro 2019

Sumário

Florestal - MG
6 de Novembro 2019

1 Introdução

O trabalho consiste em concretizar os conceitos de Análise de Complexidade de Algoritmos através da aplicação do Problema da Mochila.

O Problema da Mochila baseia-se em N itens cada um com seu respectivo peso e valor, que devem ser inseridos em uma mochila de capacidade máxima C, de forma que ela carregue o maior valor de itens o possível dentro da capacidade máxima de peso.

Para isso faremos combinações simples desses elementos, calculando todas as possibilidades e o peso e valor total dessas combinações desconsiderando as que ultrapassam a capacidade máxima da mochila, assim selecionando como a melhor aquela que possuir o maior valor total.

O desempenho do algoritmo de combinação será analisado por meio da contagem do tempo de execução em testes com um diferente número de entradas utilizando a biblioteca time.h.

Florestal - MG
6 de Novembro 2019

2 Estratégia e Organização

Primeiramente nós dedicamos os primeiros dias para entender todas as exigências requeridas para o desenvolvimento do trabalho prático. Começamos então a procurar um algoritmo de combinação simples para que pudéssemos implementar o Problema.

Dado todo o processo de entendimento das exigências, começamos o desenvolvimento a partir da criação da Lista de Itens. Primeiramente nós desenvolvemos um TAD para a lista, seguindo a implementação apresentada no livro de Nivio Ziviani[1], criando a Lista de Itens como um arranjo e também o TAD Mochila onde serão armazenadas as combinações com peso menor do que a capacidade da mochila.

Após a criação dos TADs básicos para esse sistema, seguimos as seguintes etapas:

Elaboração do menu o main para testes;
Criação da Lista de Itens e suas funções básicas;
Fazer a combinação dos itens e inseri-los na mochila.
Calcular o tempo.

Florestal - MG
6 de Novembro 2019

3 Desenvolvimento do Trabalho

Elaboração do menu no main para testes

Para facilitar a realização dos testes no programa, criamos um menu contendo uma lista de opções de operações a serem escolhidas pelo usuário, que inclui tanto a inserção de dados manualmente quanto por leitura de arquivo.

Criação da Lista de Itens e suas funções básicas

A lista de itens é uma lista por arranjo. Para a inserção na lista, inicialmente a função percorre todas as posições do vetor, inserindo cada item em uma posição. O peso e valor dos itens são recebidos no modo interativo ou por arquivo, e por isso há duas funções separadas para realizar a entrada dos dados.

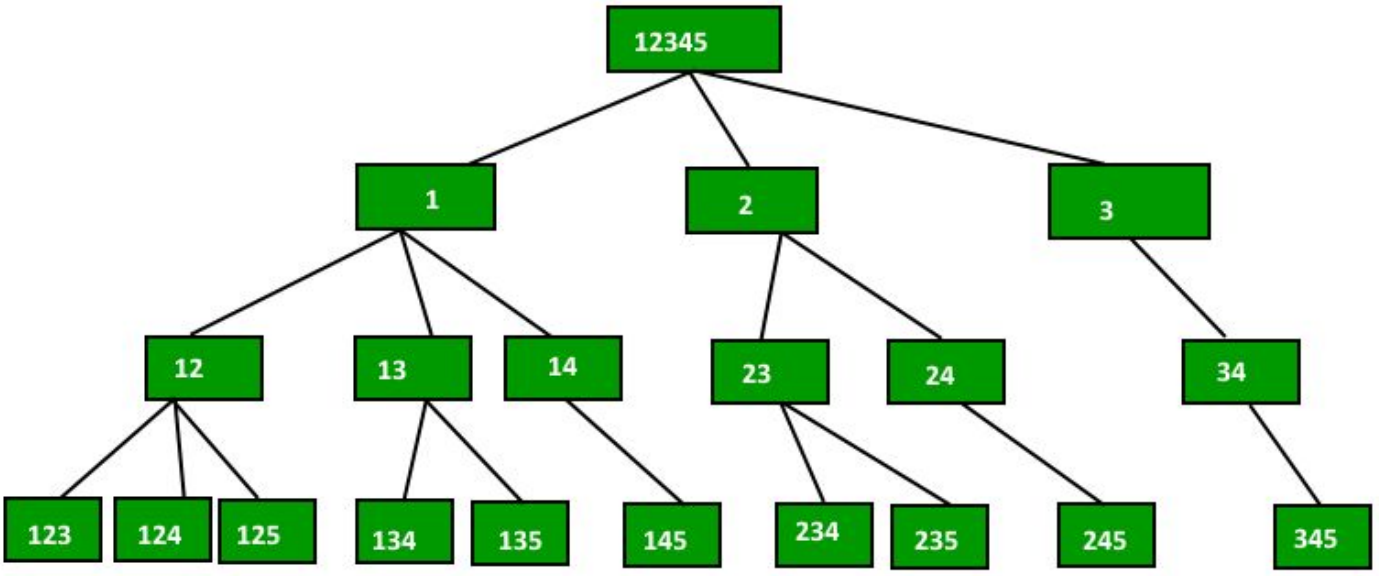
FazlistaItemVazia é uma das funções descritas por Ziviani para inicializar a lista por arranjo.
A função para imprimir a lista percorre todo o vetor imprimindo o valor e preço do objeto.

Fazer a combinação dos itens e inseri-los na mochila

O algoritmo de combinação utilizado cria um vetor temporário que vai armazenar os conjuntos de combinação. O método é começar no primeiro índice, indo um por um, inserindo os elementos neste índice, e buscando os índices restantes.

Um exemplo disso seria um vetor {1, 2, 3, 4, 5}. Colocamos o número 1 no primeiro índice e vamos buscando os restantes de acordo com a razão atual, uma vez que ela varia de 1 até n. Em seguida, fixamos o 2 no índice 0 e recorremos aos restantes. Quando o número de elementos no vetor data[ ] se torna igual a tamanho da combinação, nós printamos o vetor.

Florestal - MG
6 de Novembro 2019



Tendo em vista o objetivo do trabalho, o algoritmo[2] que nós encontramos funciona para um tamanho de combinação fixo, porém em nosso trabalho, devemos fazer tamanhos de combinação variantes de 1 a N para que seja possível atingir todas as possibilidades.

Logo após, nós avaliamos se a combinação atual é a primeira combinação possível e então armazenamos ela à mochila e para as próximas combinações, comparamos a combinação anterior - presente na mochila - e verificamos se essa combinação atual atende os requisitos para um maior aproveitamento da mochila, considerando peso e valor.

Leitura por arquivo dos itens.

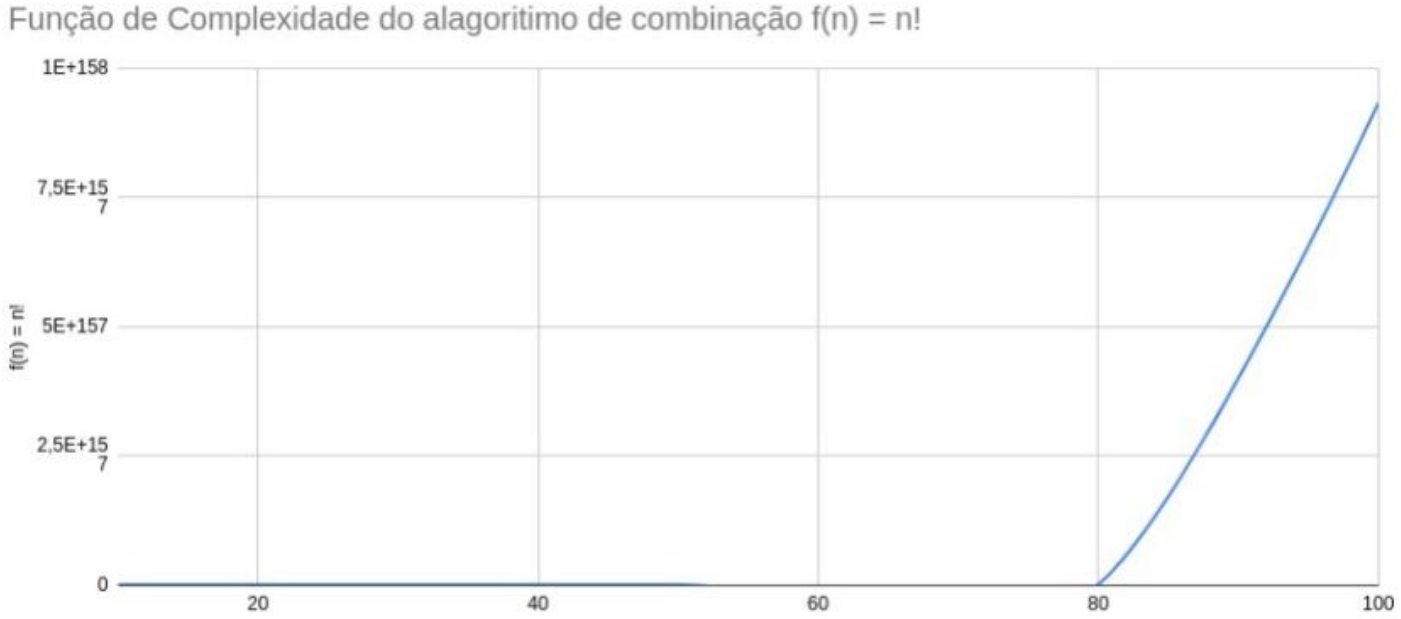
A criação da lista é feita pela leitura de um arquivo que, como a primeira linha possui apenas a quantidade de itens, usa-se o fscanf para ler essa quantidade. Em seguida, dentro da função FazListaItensArquivo utilizamos um for que se repete “quantidade” vezes, lendo o peso e valor em cada linha e salvando na lista.

Florestal - MG
6 de Novembro 2019

4 Testes

Configurações da máquina:

Processador: Intel i5-7200u
CPU: 2.5 Ghz x4
Ram: 8GB
Sistema Operacional: Deepin 15.11 Desktop



Número de Itens	Tempo de Execução
10	0,004679s
15	0,112523s
20	4,65968s
50, 80, 100	>30min.

Florestal - MG
6 de Novembro 2019

5 Conclusão

Com a execução do trabalho, percebemos a importância do uso do estudo de complexidades de algoritmos para sabermos qual é mais vantajoso de ser aplicado em determinados casos.
Como o algoritmo de combinação tem complexidade fatorial, não é razoável usá-lo para quantidades de entrada maiores do que 200, uma vez que à medida que o tamanho da entrada cresce, o tempo de execução cresce muito mais, assim, com a entrada acima de 200, o tempo gasto é muito grande, preferindo-se que em circunstâncias como essas seja abordado o uso de outro algoritmo para solucionar este problema.

Florestal - MG
6 de Novembro 2019

6 Referências Bibliográficas

1 - ZIVIANI, Nivio. Projeto de algoritmos: com implementação em Pascal e C. Cengage Learning, 2004.
[2] - Print all possible combinations of r elements in a given array of size n.
https://www.geeksforgeeks.org/print-all-possible-combinations-of-r-elements-in-a-given-array-of-size-n/

Florestal - MG
6 de Novembro 2019