

Projeto e Análise de Algoritmos

Exercício Prático 3

Bruno Schubert Trindade (121151557)

Marcos Vinícius Treviso (121150107)

1 - Análise usando Python3:

O algoritmo foi executado dividindo os valores do tamanho das músicas e da mochila por 10^3 , ou seja, a representação do tamanho foi abstraída para Kb.

É importante salientar que o algoritmo de programação dinâmica utilizado, faz com que a cada iteração m faça mais n iterações para copiar a segunda linha da matriz para a primeira. Porém mesmo assim, a complexidade continua $O(nm)$.

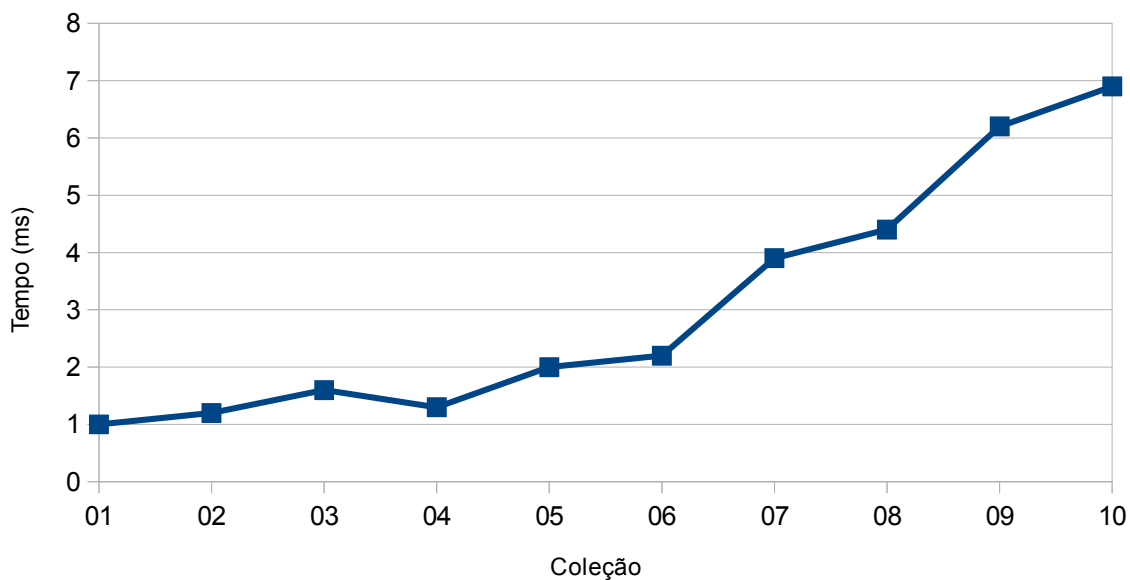
1.1 - Solução encontrada:

Algoritmo Guloso		Algoritmo programação dinâmica	
Coleção:	Resposta:	Coleção:	Resposta:
01	6	01	6
02	15	02	15
03	46	03	46
04	51	04	51
05	65	05	65
06	84	06	84
07	133	07	133
08	175	08	175
09	236	09	238
10	256	10	256

A única divergencia foi para a coleção 09, onde o algoritmo guloso não retornou a resposta esperada, já o de programação dinâmica sim.

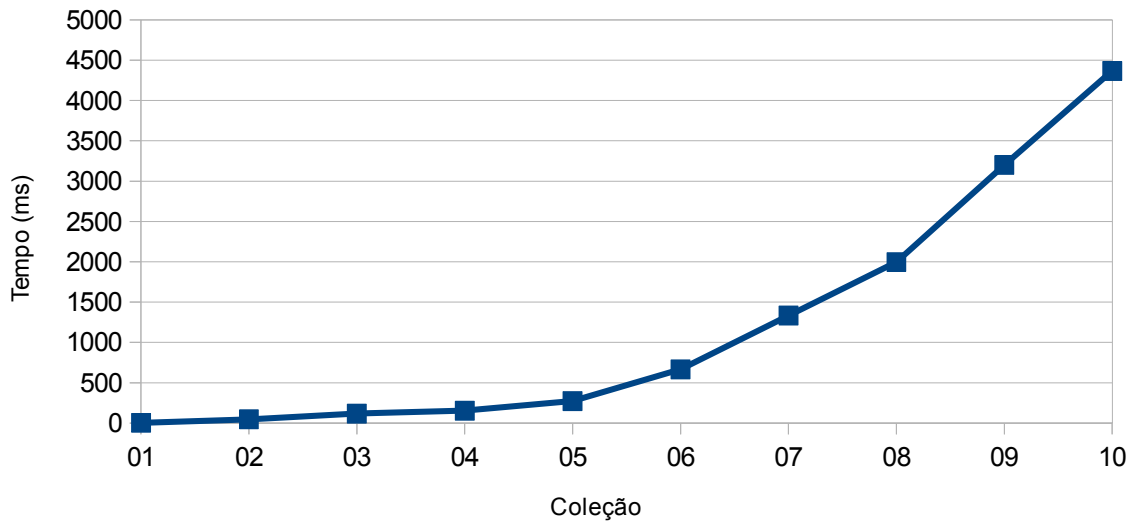
1.2 - Tempo de execução:

Algoritmo Guloso



É possível analisar através da curva obtida no gráfico que, o algoritmo possui uma complexidade linear, algo típico de um algoritmo guloso. A principal inconveniência no gráfico é quando a quarta coleção é executada, possuindo um tempo inferior a coleção anterior. Isso acontece devido ao fato que a capacidade máxima da mochila na coleção 3 é maior que a da coleção 4.

Algoritmo Programação Dinâmica



No gráfico fica visível que, de acordo com que o tamanho da mochila e quantidade de músicas aumenta, a curva tende a se estender linearmente formando a parte direita de uma parábola. Ou seja, a complexidade do algoritmo tende a se assemelhar de $O(n^2)$, entretanto, a real complexidade dele é $O(nm)$, para n igual a tamanho da mochila e m igual a quantidade de músicas à serem selecionadas.

2 – Conclusão:

O algoritmo guloso pode ser usado para questões cuja integridade da resposta não influencie no resultado final do programa, pois como visto nos gráficos, o tempo de execução dele é demasiadamente baixo se comparado ao de programação dinâmica.

Em contrapartida, o algoritmo de programação dinâmica encontra a solução correta para todos os casos, porém o tempo de execução não é muito bom. Todavia, o tempo de execução não é algo oneroso, há possibilidade de usar o algoritmo em aplicações, cujos dados usados não passem da casa dos milhões, sem que afete na velocidade da resposta.

Outro problema no algoritmo de programação dinâmica, além do tempo de execução fora do ideal, é a quantidade de memória principal utilizada para armazenar as respostas previamente obtidas, no caso do problema da mochila binária, deu para contornar tranquilamente apenas usando uma matriz com duas linhas, o que diminuiu consideravelmente a quantidade usada. Entretanto, devemos analisar bem o problema e o tipo de resposta esperada antes de escolher determinado tipo de algoritmo.

O trabalho demonstrou que uma escolha gulosa pode beneficiar em relação ao baixo tempo de execução. E o algoritmo de programação dinâmica é uma boa escolha para casos onde a integridade da resposta deve ser respeitada.