**Pontifícia Universidade Católica de Minas Gerais**

**Bacharelado em Engenharia de Software**

**Fundamentos de Projeto e Análise de Algoritmos**

**Semestre: 1/2022 - Prof. João Caram**

**Nomes**: Nataniel Geraldo Mendes Peixoto, Pedro Campos Miranda e Izabella de Castro Lucas

**Trabalho prático em grupo – parte 1**

* **Tarefas preliminares**

1. **Algoritmo força bruta:** Foi desenvolvido o algoritmo força bruta com recursividade.
2. **Método guloso:** Foi desenvolvido o algoritmo guloso.
3. **Gerador de itens:** Foi desenvolvido o método “geradorDeItens”, conforme orientações.

* **Tarefa principal**

1. **Tempo máximo execução:** Considerando a restrição do tempo, foi identificada uma instância do problema que executa em aproximadamente 2,5 segundos, dada uma capacidade de 50 e 22 itens. Se aumentar o número de itens para 23 por exemplo, a execução ultrapassa 4 segundos.

Para acompanhamento do tempo de execução, foram criadas duas listas (uma para cada algoritmo), e no final foi considerado o tempo médio de execução das 500 iterações.

1. **Teste 500 iterações:** Foi criado um teste para 500 iterações, sendo gerado um conjunto de itens distintos em cada iteração. Ressalta-se, neste caso, que o mesmo conjunto de itens foi utilizado nos dois algoritmos, para cada iteração.

Foram criadas duas listas, uma para armazenar as soluções do algoritmo por força bruta e outra para armazenar as soluções pelo método guloso. Posteriormente, foi executado um método para verificar quantas soluções foram iguais, sendo utilizado o método “equals” para cada item da lista.

Após 3 execuções, foi possível identificar que aproximadamente 51% das soluções foram iguais entre o força bruta e guloso. Ou seja, o guloso “acertou” a melhor solução em aproximadamente 51% dos casos.

Segue print dos resultados:

1º execução:



2º execução



3º execução:



O comportamento observado já era esperado.

O algoritmo força bruta gera todas as possibilidades possíveis de resposta para o problema, sendo criados vários sub-conjuntos de possibilidades. No final, o conjunto com maior valor e peso (que não ultrapasse a capacidade) é a melhor solução.

O tempo gasto para encontrar a melhor solução é exponencial. Por isso, para uma grande quantidade de dados, este algoritmo não é eficiente.

Já o método guloso, pode ou não retornar a melhor solução, porém ele executa de forma mais rápida. O tempo de execução é da ordem de n.

* **Pontos extras**

Foram realizados vários testes com capacidades crescentes de mochila, basicamente dobrando seu valor. Em todos os testes, conforme é possível verificar abaixo, o tempo médio de execução do algoritmo força bruta não variou de forma significativa, nem mesmo a quantidade de soluções iguais entre Força Bruta x Guloso. Vale destacar que não foi alterada em nenhum caso o número de itens, permanecendo em 22.

Este comportamento pode ser explicado pelo fato de a geração de pesos aleatórios ter sido desenvolvida de forma a ser de 3x a capacidade da mochila. Sendo assim, mesmo duplicando a capacidade da mochila, a distribuição aleatória dos pesos teve o aumento proporcional. Então a lógica básica de execução do programa não foi alterada.

Capacidade: 100



Capacidade: 200



Capacidade: 400



Capacidade: 800



Capacidade: 1.600



Capacidade: 3.200



Capacidade: 10.000

