

Projeto prático 4

SCC 0606 – Estrutura de Dados II
Prof. Maria Cristina Ferreira de Oliveira
10 de junho de 2024

1. Descrição

Júlia está cursando a matéria de Estruturas de Dados na faculdade, e está aprendendo sobre os diferentes paradigmas de programação. Um dos paradigmas que achou mais interessante foi o dos algoritmos gulosos, que é altamente utilizado para alguns tipos de problemas, apesar de não necessariamente encontrar o melhor resultado possível. Em particular, Júlia ficou curiosa para entender melhor quão piores podem ser, na prática, os resultados encontrados por um algoritmo guloso quando comparado com a solução ótima. Ela quer analisar melhor essa questão, e espera contar com a sua ajuda para isso. Sua tarefa é desenvolver um programa que permita analisar as soluções obtidas para um problema específico.

O problema a ser resolvido será o Problema da Mochila. Dada uma entrada definindo o peso máximo suportado pela mochila e os pesos e os valores de diferentes itens, para ajudar a Júlia você precisa: (i) utilizar programação dinâmica para identificar a solução ótima, i.e., os itens a serem carregados que correspondem ao valor máximo possível; (ii) construir um algoritmo guloso que seleciona os itens por ordem de prioridade da razão entre o valor e peso (em caso de empate, dar preferência para o item com menor peso), e para de procurar itens válidos ao encontrar o primeiro item que não cabe na mochila; e (iii) comparar os resultados.

2. Entrada e Saída

Primeiramente, seu programa receberá na entrada P e Q , que informam, respectivamente, o peso máximo que a mochila aguenta, e a quantidade de itens a serem informados. Em seguida, uma sequência de pares de inteiros, que representam, respectivamente, o peso e o valor de cada item.

Na primeira linha, o programa deverá imprimir quais foram os itens selecionados na solução ótima (passando o índice do item, baseado na ordem dos itens passados na entrada, sendo o primeiro item o item 0, e em ordem numérica, ou seja, do item de menor número para o de maior número selecionado), seguido (na segunda linha) pelos itens selecionados pelo algoritmo guloso (mesma lógica na identificação numérica dos itens, todavia ao invés de ordem numérica deve ser passada a ordem de seleção dos itens, ou seja, qual é o mais prioritário).

Na terceira linha deverá ser impresso o valor total gerado pela programação dinâmica, seguido do valor total gerado pelo algoritmo guloso, e por fim, na quarta linha, a porcentagem do valor obtido pelo algoritmo guloso, quando comparado ao

valor ótimo (use valores com precisão de até 2 casas após a vírgula, arredondado para o valor mais próximo).

3. Conteúdo e data de entrega

O trabalho é individual e a data limite de submissão é **30/06/2024** (trabalhos em atraso não serão aceitos), no sistema RunCodes (<https://runcodes.icmc.usp.br/>). O formato da entrega deve ser um arquivo com o código fonte desenvolvido.

- O código deverá estar adequadamente comentado/documentado;
- O trabalho pode ser desenvolvido utilizando C ou Python, a escolha;
- Não é permitido utilizar bibliotecas que já disponibilizam os algoritmos de forma parcial ou total, todavia, **poderão ser utilizadas bibliotecas para a ordenação** por prioridade necessária no algoritmo guloso;
- Todo o código deve ser desenvolvido pelo próprio aluno. Os códigos serão submetidos a um detector de plágio, e a utilização de código criado por terceiros será tratada como plágio (nota zero).

Dica:

(i) caso seu programa esteja gerando alguma saída errada, mas está difícil identificar a diferença entre a saída esperada e a resultante, este site para comparação de textos pode ser útil: <https://www.diffchecker.com/text-compare/>

4. Critérios de avaliação

- 9,0 pontos pela implementação dos algoritmos solicitados, de forma correta (avaliado segundo o resultado do RunCodes, e por inspeção do código).
- 1,0 ponto por código legível, bem estruturado e bem comentado.

5. Exemplos de entrada e saída

Segue abaixo um exemplo de entrada, lógica e saída esperadas:

Entrada	Lógica	Saída esperada
8 4 2 1 3 2 4 5 5 6	Como este é um caso pequeno, podemos facilmente concluir que o caso ideal (a ser gerado pela programação dinâmica) é a escolha dos itens 1 (3 de peso, 2 de valor) e 3 (5 de peso, 6 de valor), uma vez que estes geram o maior valor possível (8 de valor) para uma mochila que comporta 8 de peso.	1 3 2 8 5 62.50

	<p>A lógica de como gerar tal resultado usando programação dinâmica deverá ser formulada por vocês.</p> <p>Para o algoritmo guloso, basta ordenar os valores por ordem de prioridade (razão entre valor e peso):</p> <ul style="list-style-type: none">• Item 2: Peso 4, valor 5, razão 1.25• Item 3: Peso 5, valor 6, razão 1.2• Item 1: Peso 3, valor 2, razão 0.66• Item 0: Peso 2, valor 1, razão 0.5 <p>Assim sendo, começamos escolhendo pelo item 2, e vamos adicionando itens até não ser possível adicionar o próximo item da lista. Neste caso, o programa já para no item 3, uma vez que o peso dos itens 2 e 3 já ultrapassam o peso da mochila. Logo, apenas o item 2 é selecionado.</p>	
--	--	--