



Implementação Algorítmica

Trabalho Avaliativo 2

Sophya Martins Ribeiro
Soraya Dias Ferreira

INTRODUÇÃO

O presente relatório visa expor análises acerca de resultados dos testes de precisão e tempo de execução realizados com dois algoritmos de resolução do problema Corte da Tora, para a disciplina de Implementação Algorítmica. Nos testes, foram comparadas uma solução em programação dinâmica e outra com a estratégia gulosa. Todas as análises e os algoritmos foram desenvolvidos utilizando a linguagem Python com algumas bibliotecas, tais como Time — para leitura dos tempos iniciais e finais de execução —, Matplotlib — para gerar gráficos baseados nos dados obtidos —, numpy — para melhor visualização do gráfico — e random — direcionada aos números aleatórios da lista de preços.

Em todos os testes realizados é necessário considerar três parâmetros principais: INC e FIM — que referem-se consecutivamente à posição inicial e final para os vetores utilizados e, por fim, STP — que denota o passo, ou intervalo, em que serão registrados os tempos de execução.

O problema do corte da tora consiste em encontrar a melhor maneira de cortar uma tora de madeira em pedaços menores, de modo a maximizar o valor total obtido com a venda desses pedaços. Cada pedaço tem um valor associado, e a tarefa é determinar onde fazer os cortes para alcançar o maior retorno possível. Neste trabalho foram considerados dois algoritmos para resolver esse problema: um utilizando programação dinâmica *bottom up*, que examina todas as opções possíveis para garantir a solução ótima; outro fazendo uso da abordagem gulosa, que toma decisões localmente ótimas, isto é, escolhendo sempre os cortes com maior densidade, o que não garante que o resultado final será o melhor possível. Esses algoritmos são comparados tanto em termos de tempo de execução quanto no valor total obtido com a venda das toras cortadas.



INSTRUÇÕES PARA EXECUTAR O PROGRAMA

Para executar os programas, é necessário que a máquina tenha o Python 3 instalado e seja instalada a biblioteca matplotlib por meio do comando "pip install -U matplotlib", conforme documentação da ferramenta. Tendo isso, basta escolher um ambiente de desenvolvimento integrado e abrir a pasta do projeto. Para realizar os testes considerando os parâmetros explicados anteriormente, basta rodar o arquivo main.py e inserir as entradas solicitadas no terminal.

ALGORITMOS

Os vetores de preços foram gerados aleatoriamente por meio da função "gerar_precos(n: int) -> list", que cria uma lista de tamanho n , na qual cada elemento é um número inteiro positivo aleatório, gerado pela função `random.randint(1, n)`. Essa abordagem permite gerar valores dentro do intervalo $[1, n]$. O valor superior variou em n , $2*n$, $3*n$ ou até $4*n$ para tentar obter maior variação entre as soluções.

A solução dinâmica consiste em aplicar o método *bottom-up* da programação dinâmica para encontrar o valor máximo possível de venda. A ideia é construir uma tabela de soluções parciais de forma incremental, armazenando os melhores resultados de cortes já realizados e reutilizando-os para calcular cortes maiores. No código, temos uma lista chamada *storage*, que guarda os valores máximos obtidos para cada tamanho de tora, desde o tamanho 0 até o tamanho total da tora. Para cada comprimento, a solução mais lucrativa é calculada testando diferentes pontos de corte e somando o valor do pedaço cortado ao valor já otimizado do restante. Ao final, a tabela contém a melhor solução para o tamanho total da tora, que é o valor retornado. Com isso, garantimos que todas as combinações possíveis de cortes são consideradas, e o valor máximo de revenda é obtido de forma eficiente.

Já a solução gulosa que usamos busca sempre a tora que tem o melhor custo-benefício em relação ao tamanho. Em vez de calcular todas as possibilidades, ela vai direto na escolha que parece mais vantajosa naquele momento, comparando o valor por unidade de tamanho das toras e considerando aquela que traz mais retorno por centímetro. Diferente da solução dinâmica, que considera todas as combinações de cortes possíveis, o algoritmo guloso não garante que a solução encontrada será a máxima, mas tenta obter um valor alto com base em decisões imediatas. É uma abordagem mais simples e rápida, mas pode resultar em um valor de revenda inferior ao da solução dinâmica, que otimiza globalmente.



RESULTADOS OBTIDOS

Nesta seção serão apresentados os resultados obtidos para cada parâmetro escolhido, considerando o gráfico de tempo de execução entre as duas soluções e o gráfico dos valores de quantidade máxima de venda obtidos. Para o primeiro teste, foram considerados os parâmetros explícitos na tabela 1, para o segundo teste, os da tabela 2 e assim por diante. A linha alaranjada do gráfico com nome Greedy refere-se ao algoritmo guloso, enquanto que a linha azul com nome Dynamic Programming refere-se à solução dinâmica.

Tabela 1 - Parâmetros para realização do teste 1

Teste 1			
Parâmetros	INC	FIM	STP
Valores	1000	20000	1000

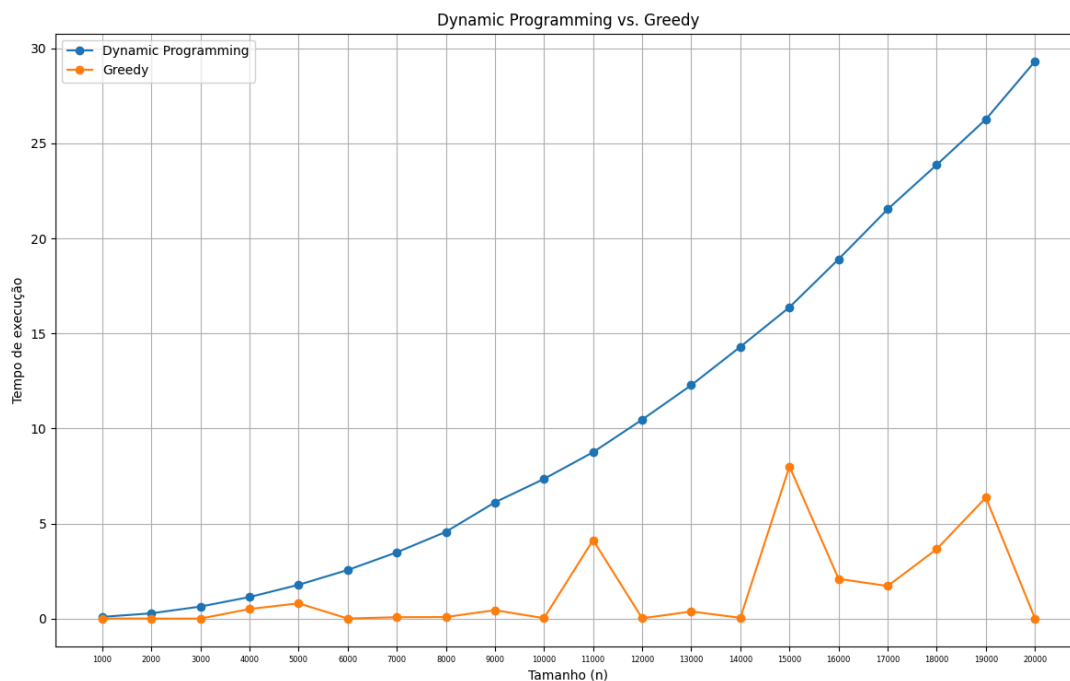
Fonte: autoras, 2024.

Como pode ser observado no Gráfico 1 abaixo, o algoritmo guloso é mais eficiente que o dinâmico. No entanto, a despeito disso, ele não pode ser considerado melhor, uma vez que seu resultado não é 100% correto em todos os casos, ou seja, ele é ineficaz.

É importante explicitar que, no Gráfico 2, que demonstra a comparação dos totais de venda, pode parecer que somente o Dynamic Programming foi representado, todavia isso se deve ao fato de que o Greedy errou raramente e, quando errou, teve uma porcentagem de acerto sempre igual ou superior a 99%. o que dificulta a visualização gráfica das discrepâncias.

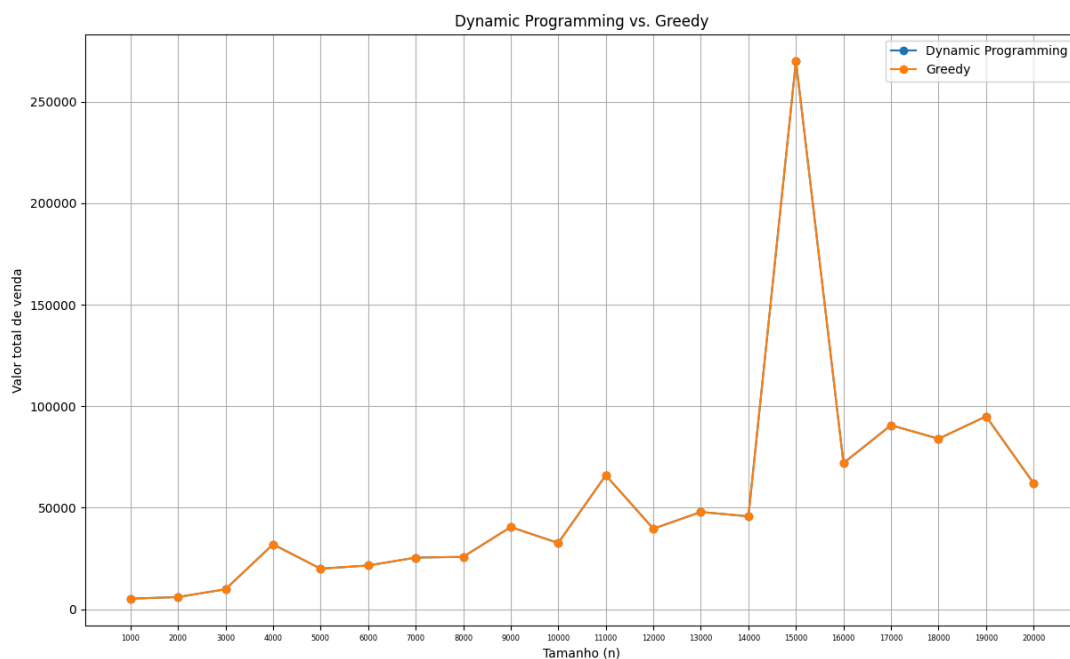


Gráfico 1 - Tempos obtidos no teste 1



Fonte: autoras, 2024.

Gráfico 2 - Quantidade de vendas obtidas no teste 1



Fonte: autoras, 2024.



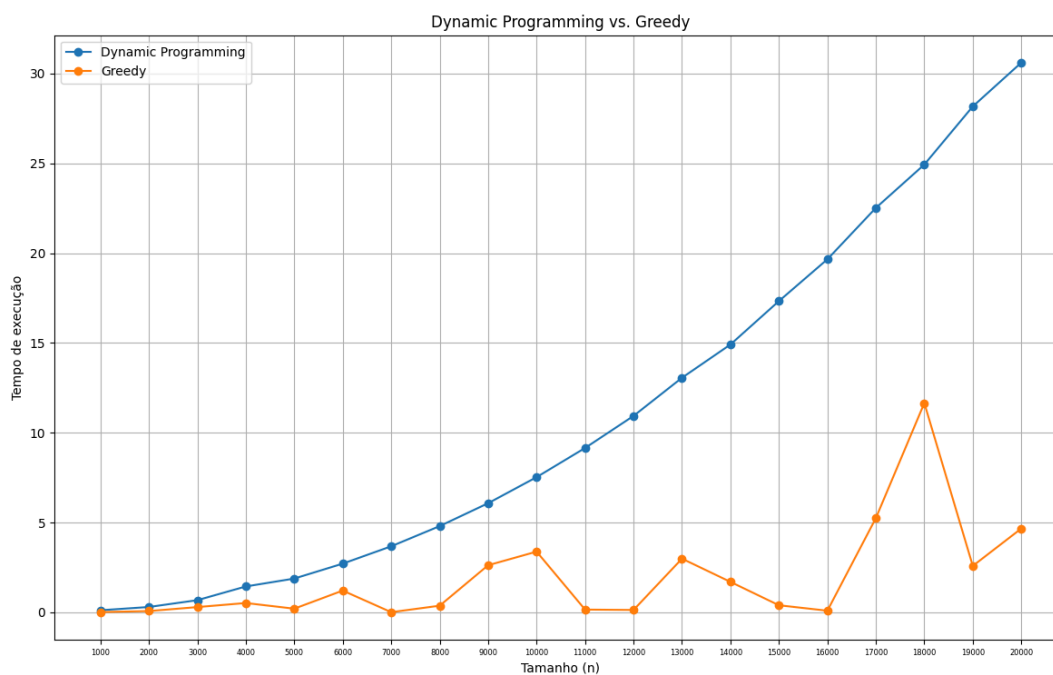
Tabela 2 - Parâmetros para realização do teste 2

Teste 2			
Parâmetros	INC	FIM	STP
Valores	1000	10000	1000

Fonte: autoras, 2024.

Nos próximos testes, conforme os parâmetros aumentaram, o comportamento de ambas as operações prosseguiram seguindo o que era previsto.

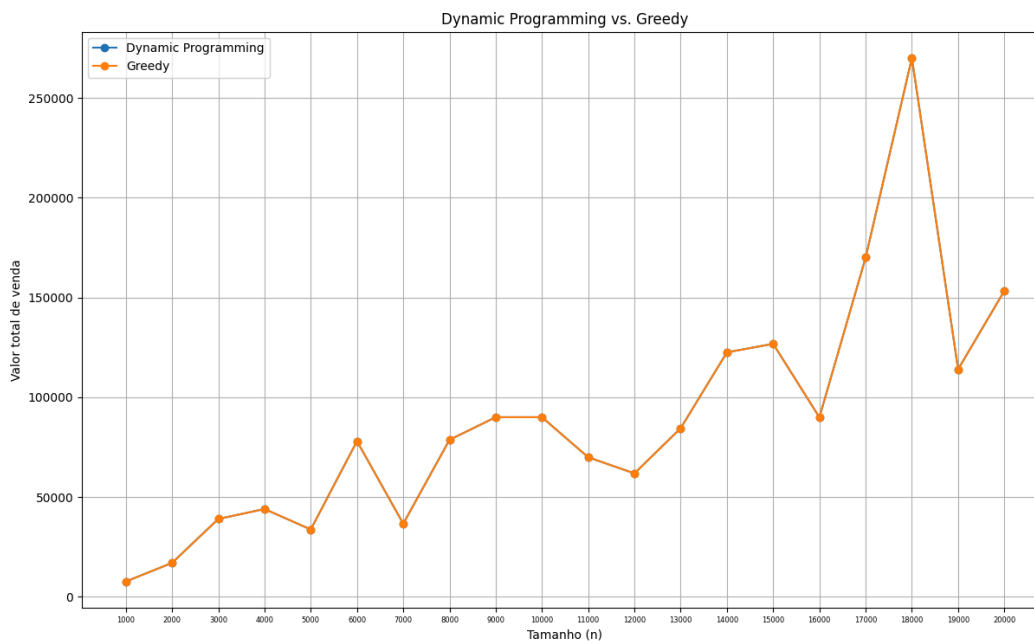
Gráfico 3 - Tempos obtidos no teste 2



Fonte: autoras, 2024.



Gráfico 4 - Quantidade de vendas obtidas no teste 2



Fonte: autoras, 2024.

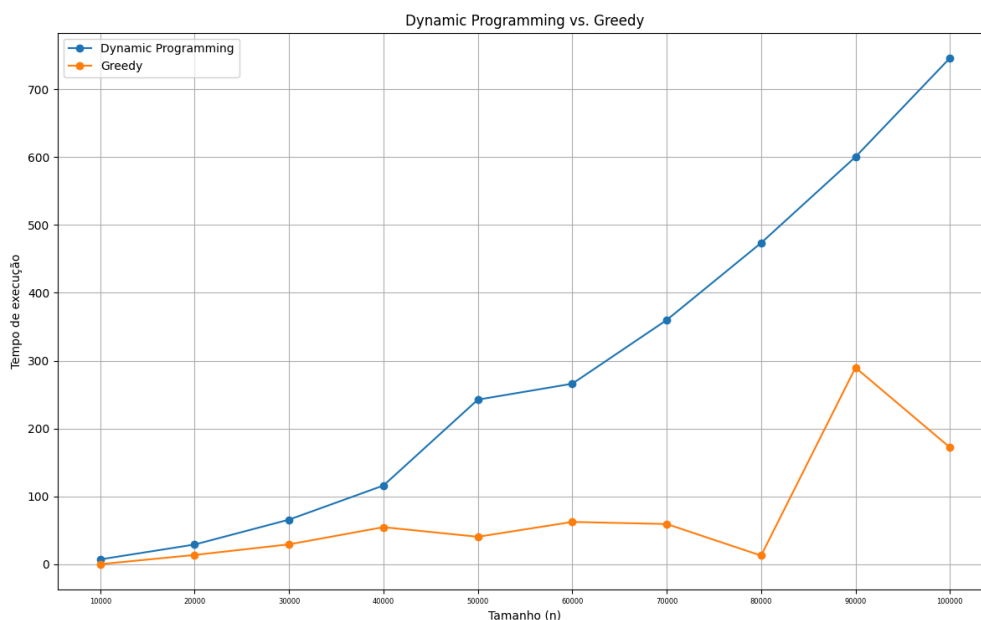
Tabela 3 - Parâmetros para realização do teste 3

Teste 3			
Parâmetros	INC	FIM	STP
Valores	10000	100000	10000

Fonte: autoras, 2024.

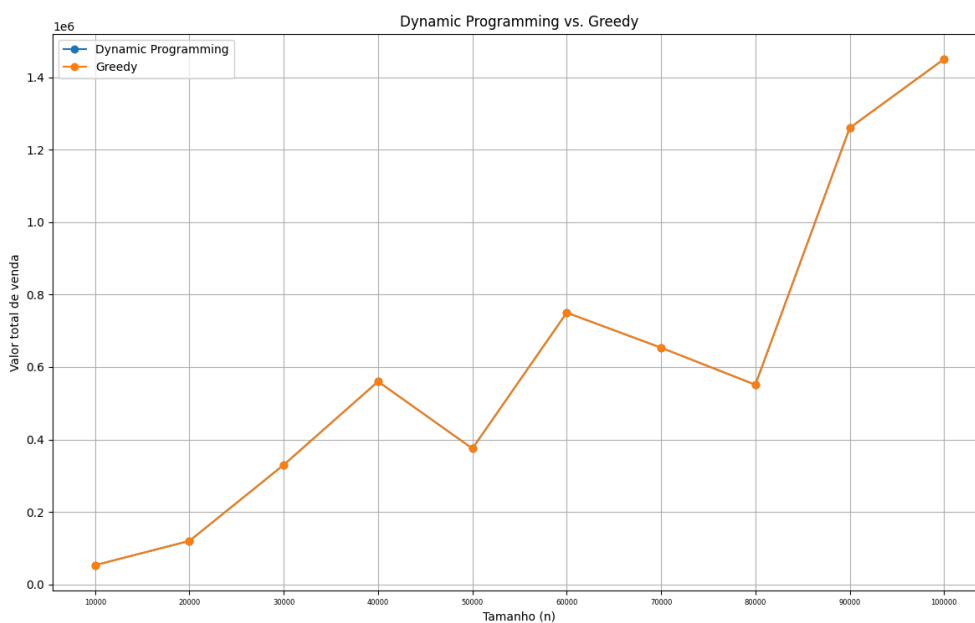


Gráfico 5 - Tempos obtidos no teste 3



Fonte: autoras, 2024.

Gráfico 6 - Quantidade de vendas obtidas no teste 3



Fonte: autoras, 2024.



Serviço Público Federal
Ministério da Educação

Fundação Universidade Federal de Mato Grosso do Sul



CONSIDERAÇÕES FINAIS

Neste relatório, foram analisados os tempos de execução e os resultados esperados quanto ao valor total de vendas do problema do corte da tora utilizando uma solução gulosa e uma solução dinâmica. Os resultados saíram de acordo com o esperado. A solução gulosa, porém, surpreendeu, pois foi capaz de encontrar a exata solução em quase todos os casos de teste, e a porcentagem de aproximação dos totais de venda dela não esteve abaixo de 99% em nenhum momento.



Serviço Público Federal
Ministério da Educação

Fundação Universidade Federal de Mato Grosso do Sul



REFERÊNCIAS

MATPLOTLIB. *User guide*. Disponível em: <https://matplotlib.org/stable/users/index>. Acesso em: 05 set. 2024.

HIGA, Carlos. Slides da Disciplina. Apresentações. Disponível em: <https://ava.ufms.br/course/view.php?id=61684>. Acesso em 08 set. 2024.