

1. Definições

1.1 Algoritmo Guloso: São comumente utilizados para resolver problemas de otimização por serem geralmente, de simples implementação. Algoritmos gulosos são uma maneira de gerar solução ótima para um problema pois escolhem um foco e geram uma solução baseada nisso. Dessa maneira, o algoritmo sempre escolhe a solução ótima local baseada no foco escolhido, com o objetivo de juntar suas soluções ótimas locais e gerar uma ótima global. Infelizmente, dependendo de qual for o foco escolhido, o algoritmo guloso pode não gerar solução ótima, sendo assim, uma parte importante desta técnica é saber escolher bem qual o foco que será utilizado como base para as escolhas a serem realizadas.

1.2 Programação Dinâmica: São muito utilizadas na área da computação para resolver problemas de otimização combinatória por sempre gerar solução ótima para o problema dado. Algoritmos que utilizam Programação Dinâmica resolvem problemas cuidando dos subproblemas primeiro para depois lidar com o problema principal. Sendo assim, é montada uma tabela com as soluções ótimas para cada subproblema encontrado, e ao preencher a tabela por completo, é encontrada a solução ótima para o problema maior. Dessa maneira, por ser criada uma tabela que pode ser encarada como uma memória interna do algoritmo, uma mesma situação não precisa ser calculada mais de uma vez, o que pode otimizar códigos com múltiplas entradas de teste.

2. Perguntas

2.1 Como esse problema pode ser modelado para o paradigma guloso?

O problema pode ser modelado utilizando a média entre preço e custo como foco, sendo que diariamente a média entre preço e custo de todos os pratos é analisada e o que obtiver a maior média é escolhido para o dia. Sendo assim, tendo sido feita a escolha do prato, uma nova média é calculada para o prato escolhido baseando-se no fato de que, se ele for escolhido novamente em sequência, o lucro deve ser apenas 50% do lucro original do prato. Se um mesmo prato for escolhido 3 vezes seguidas sua média passa a ser 0, e se um prato diferente for escolhido, a antiga média alterada do prato deve ser retornada à média original.

2.2 Seu algoritmo guloso dá a solução ótima? Por quê?

Não, pois a solução gulosa nesse problema pega o melhor valor no momento atual, ou seja, o cozinheiro irá cozinhar o prato que lhe dará o melhor lucro no dia em questão sem levar em conta todos os dias que ele tem disponíveis ou o orçamento total que ele possui. Dessa forma, o algoritmo guloso não garante a solução ótima para o problema como um todo, e sim para o problema nos dias em questão.

2.3 Como esse problema pode ser modelado para o paradigma de programação dinâmica?

Para modelar o problema em maneira dinâmica, utilizaremos n tabelas (quantidade de dias), que serão compostas pelo orçamento atual e referência dos pratos existentes para um determinado problema. A tabela seria preenchida de forma que, iremos guardar o maior valor lucro obtido no dia, analisando os pratos existentes, semelhante ao problema da mochila, mas diferenciado em um quesito que, se só se escolhe um prato por dia, logo o maior orçamento irá repetir em todo o restante da tabela, até encontrar um outro maior, ou até o maior, caso contrário.

Juntamente a isso, teremos uma análise dos pratos utilizados, e verificaremos os pratos que serão utilizados se serão repetidos ou não, por meio de uma tabela auxiliar, que teria a mesma ideia do que a primeira, só que teria referência do item que fora utilizado no dia anterior, e caso não tenha nenhum no dia anterior, ou não puder ser pego nenhum deles, teria o valor de -1, contido em sua posição. Essa tabela também auxiliaria na construção do caminho que levaria a melhor solução.

A saída teria a mesma forma que o algoritmo guloso. Entretanto, para a descoberta do mesmo, verificaremos as condições foram atendidas para a entrada (o orçamento, pratos foi suficiente para todos os dias). Caso as condições for verdadeira, iremos verificar a primeira tabela para verificar qual o maior obtido em todos os dias de preparação, e analisaremos a tabela auxiliar para verificar quais foram os pratos utilizados para tal conclusão, onde para isso, iremos pegar o valor atual da tabela 1, verificar a mesma posição, porém na tabela auxiliar e verificar o prato anterior e assim por diante. E, se as condições forem falsas, iremos mostrar a impossibilidade da solução.

2.4 Discuta a subestrutura ótima e a sobreposição dos problemas.

O algoritmo utiliza uma cadeia de laços de repetição para trabalhar com os vetores de tabelas, sendo que para cada dia existe uma tabela de lucro, e duas de memória, as de memória guardam as frequências de cada prato utilizado no menu e cada um dos pratos utilizados em cada dia. Para resolver o problema o algoritmo trabalha como qualquer outro algoritmo Dinâmico, ele resolve os subproblemas para resolver o problema maior, sendo assim, o menu é calculado para todos os orçamentos possíveis antes de ser calculado para o orçamento dado, por exemplo, se o orçamento dado é o máximo possível (100), ao preencher as tabelas, é encontrado a solução ótima para o orçamento de 1, de 2 de 3 e assim por diante, até chegar na última coluna da tabela cujo orçamento é 100 (no caso), tendo chegado a este ponto, os menus gerados para os subproblemas são utilizados como base para criar o menu ótimo para o problema maior.

2.5 Se algum algoritmo clássico foi adaptado para resolver o problema, qual foi ele?

O Algoritmo clássico adaptado para resolver o problema do Trabalho Prático, foi o algoritmo da mochila binária, onde o objetivo é maximizar o valor total dos itens que podem ser carregados na mochila. Dessa forma, foi possível adaptá-lo para um problema em que deve-se maximizar o lucro ao longo dos dias levando em consideração as regras impostas pelo problema.

2.6 Qual o custo assintótico (em Big-O) do tempo de execução e espaço em memória utilizado? Não se esqueça de formular a equação de recorrência da abordagem baseada em programação dinâmica (não é necessário resolver a equação explicitamente).

O custo assintótico do tempo de execução do algoritmo Guloso, o qual apresenta duas estruturas principais, um laço de repetição externo que é executado “n” vezes (sendo n o número de dias informado) e um laço mais interno que busca a maior média no vetor de pratos que é de tamanho “m”. Sendo assim, assintoticamente, o algoritmo guloso apresenta um custo $O(mn)$.

O Dinâmico apresenta cinco estruturas principais, um laço no primeiro nível, que é executado “n” vezes, um laço no segundo nível, que é executado “m” vezes, um laço no terceiro nível, que é executado “k” vezes, um laço no quarto nível que é executado “ $\sum i$ até n, sendo $i = 0$ ” vezes e um laço no quinto nível que é executado “m” vezes também sendo assim, temos a equação:

$$n * m * k + [m * k * (\sum i \text{ até } n, \text{ sendo } i = 0) * m] =$$

$$n * m * k + [m^2 * K * (n * (n+1)) / 2] =$$

$$n * m * k + (m^2 n^2 + m^2 n) / 2 =$$

$$(2nmk + m^2 n^2 + m^2 n) / 2$$

o algoritmo utilizando Programação Dinâmica apresenta custo $O(n^3)$ para o melhor caso, e no pior caso apresenta custo $O(m^2 n^2)$.

3. Máquina

produto: Intel(R) Core(TM) i7-7820HK CPU @ 2.90GHz

ram: 16GB

tipo: 64 bits

capacidade: 500GB disco local(C:), 1TB disco local(D:)

sistema operacional: Windows 10

Códigos executados no NetBeans.

4. Bibliografia

Paulo Feofiloff. Análise de algoritmos, 2015 http://www.ime.usp.br/~pf/analise_de_algoritmos/

CORMEN, Thomas H. et al. Algoritmos: teoria e prática. Rio de Janeiro, RJ: Elsevier, Campus, c2012. xvi, 926 p. ISBN 9788535236996., Nº de Exemplares: 14.

Felipe Cunha. Notas de Aula - Técnicas de Projeto, 2018

