Nome: Wayne Nascimento Souza
Matrícula: 0016183

Paradigma da Programação Dinâmica

A Programação Dinâmica é uma metodologia de desenvolvimento que divide problemas complexos em questões mais simples e os resolve repetidamente. Segundo Ferreira e Matheus (2021) a principal inovação dessa abordagem é o uso de memória para armazenar os subresultados, que são reutilizados ao longo do cálculo da solução geral em diferentes momentos.

Conforme o site Geeksforgeeks (2022), a subestrutura ótima e a sobreposição de soluçõessão as bases desta abordagem. A subestrutura ótima diz que a melhor solução para um problema é composta pelas melhores soluções para seus subproblemas dependentes. A sobreposição de soluções mostra que a resolução de subproblemas que surgem duas ou mais vezes é a solução ideal.

De acordo com Feofiloff (2020) a Programação Dinâmica oferece uma maneira eficaz de resolver problemas, usando uma tabela que armazena as soluções das várias subinstâncias. O tamanho da tabela geralmente corresponde ao tempo gasto pelo algoritmo.

O problema deve ter uma estrutura recursiva, com soluções de subinstâncias para cada instância, para que o paradigma em questão possa ser usado, afirma Kleina (2021). A Programação Dinâmica também é útil quando subproblemas são repetidos, pois as soluções calculadas podem ser armazenadas e reutilizadas, evitando cálculos inúteis. Segundo o site Geeksforgeeks (2023), para acelerar os programas de computador, a memoização é uma estratégia que elimina computações repetitivas e evita chamadas repetidas de funções que processam a mesma entrada. Como resultado, a memoização e a programação dinâmica são ferramentas poderosas para lidar com problemas complexos e otimizar o desempenho de algoritmos, especialmente quando vários problemas se sobrepõem.

Geralmente tem a ver com encontrar o máximo e o mínimo intervalo da consulta algorítmica, declara BasuMallick (2022), e alguns exemplos de problema clássicos que podem ser solucionados usando a Programação Dinâmica segundo o site Medium (2018) são: Subsequência comum mais curta, subsequência comum mais longa, multiplicação de cadeias de matrizes, problema binário da mochila entre outros.

Modelagem da solução

Para realizar a modelagem de uma possível solução para o problema, primeiramente foi definido com base no enunciado de como seriam a entrada e a saída. A entrada é definida pelo número de itens na lista da Senhora Jones, o número total de produtos disponíveis no supermercado, uma lista com os produtos da lista da Senhora Jones e uma série de valores composta pelo identificador do produto e o seu preço. Já a saída, retorna o menor custo que o Senhor Jones pode conseguir. Com isso, foi possível determinar o formato dos candidatos e desenvolver um algoritmo baseado no paradigma da Programação Dinâmica para solucionar o problema do supermercado (1351) presente na plataforma Beecrowd.

A aplicação do paradigma da Programação Dinâmica como parte da solução possui as seguintes etapas: identificação do problema, definição da entrada e saída, definição da estrutura de tabela, preenchimento da tabela, cálculo dos valores das células e verificação da solução final.

Na primeira etapa, é necessário identificar se o problema pode ser resolvido usando este paradigma. Como o objetivo deste problema é a minimização do custo total da compra em um supermercado, então é possível utilizar.

A etapa de definição da entrada e saída, para solucionar o problema é utilizada a leitura de entrada para obter o número de itens da lista da Senhora Jones (M), o número de total de produtos disponíveis no supermercado (N), a lista de produtos da Senhora Jones (jones_list) e a lista de preços dos produtos nos supermercados (supermarket_list). Já a saída é o menor custo caso seja possível, se não é retornado Impossible.

A definição da estrutura de tabela é a terceira etapa, onde é usada uma matriz para armazenar as soluções dos subproblemas e no ‘sentido’ da diagonal principal é representado a cada item o custo total dos produtos escolhidos pelo Senhor Jones.

Na etapa do preenchimento da tabela é onde ocorre a criação da tabela com valores que simbolizam o infinito e logo após a primeira linha da matriz é preenchida com zero porque até então nenhum produto foi selecionado.

A próxima etapa é para realizar o cálculo dos valores das células da matriz através de relações de recorrência do problema. Isso é feito com duas estruturas de repetição aninhadas e para cada célula é verificado se o produto da lista da Senhora Jones está disponível na lista de produtos do supermercado, caso esteja disponível o custo mínimo é calculado escolhendo o menor valor entre comprar o produto e não comprar e, caso não esteja disponível o custo mínimo é copiado da célula anterior na mesma linha.

E por fim, a etapa da verificação da solução final. Após preencher a tabela é verificado se a célula da matriz na posição [M][N] possui o valor usado para simbolizar o infinito, caso seja, então significa que não é possível realizar a compra solicitada pela Senhora Jones e é

retornado Impossible. Caso contrário, o custo mínimo é retornado como resultado.
Com base nessa modelagem, foi possível resolver o problema 1351 do Beecrowd. Logo abaixo estão algumas evidências que o código oriundo desta modelagem foi aceito pela bateria de testes e no tempo definido pela plataforma.

34291738	1351	Supermercado	Accepted	Python 3.8	3.137	24/06/2023 18:08

SUBMISSÃO # 34291738

PROBLEMA:	1351 - Supermercado
RESPOSTA:	Accepted
LINGUAGEM:	Python 3.8 (Python 3.8.2) [+1s]
TEMPO:	3.137s
TAMANHO:	1,72 KB
MEMÓRIA:	-
SUBMISSÃO:	24/06/2023 18:08:22

Análise do custo do algoritmo

Para analisar o custo do algoritmo será dividido em trechos de códigos, apontando a linha inicial e a final e consequentemente o custo associado.

- 07: é a inicialização da matriz (M * N).
- 10-15: inicialização e atribuição da lista com os produtos (1 + N).
- 18-28: preenchimento da tabela de custos mínimos, verificação da disponibilidade do produto, cálculo do custo mínimo entre efetuar a compra ou não e atribuição do custo mínimo (M * N + 1 + 1).
- 31-34: verificação da possibilidade de comprar todos os produtos (1).

A soma de todos esses custos resulta na equação recorrência abaixo:
$T(n) = O(m * n) + O(1) + O(n) + O(n) + O(1) + O(1) + O(1)$
Como o n é a complexidade assintótica dominante dentre estas, podemos usar a equação simplificada $T(n) = 2 . O(m * n)$.

Não é possível aplicar o Teorema Mestre pois não temos a equação de recorrência na forma $T(n) = aT(\frac{n}{b}) + f(n)$.

Logo vamos considerar que o custo deste algoritmo é $O(m * n)$.

Referências

MEDIUM. Top 10 Dynamic programming problems for interviews, 2018. Disponível em: https://medium.com/techie-delight/top-10-dynamic-programming-problems-5da486eeb360. Acesso em: 25 de jun. de 2023.

FERREIRA, Ennio; MATHEUS, Victor. Programação Dinâmica, 2021. Disponível em: https://lamfo.unb.br/wp-content/uploads/2021/03/Programa%C3%A7%C3%A3o-Din%C3%A2mica.pdf. Acesso em: 25 de jun. de 2023.

FEOFILOFF, Paulo. Programação Dinâmica, 2020. Disponível em: https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/dynamic-programming.html. Acesso em: 25 de jun. de 2023.

KLEINA, Mariana. Programação Dinâmica, 2021. Disponível em: https://docs.ufpr.br/~marianakleina/PD-aula2-exemplos.pdf. Acesso em: 25 de jun. de 2023.

GEEKSFORGEEKS. Overlapping Subproblems Property in Dynamic Programming, 2022. Disponível em: https://www.geeksforgeeks.org/overlapping-subproblems-property-in-dynamic-programming-dp-1. Acesso em: 25 de jun. de 2023.

GEEKSFORGEEKS. Optimal Substructure Property in Dynamic Programming, 2022. Disponível em: https://www.geeksforgeeks.org/optimal-substructure-property-in-dynamic-programming-dp-2. Acesso em: 25 de jun. de 2023.

GEEKSFORGEEKS. What is memoization? A Complete tutorial, 2023. Disponível em: https://www.geeksforgeeks.org/what-is-memoization-a-complete-tutorial. Acesso em: 25 de jun. de 2023.

BASUMALLICK, Chiradeep. What is Dynamic Programming? Working, Algorithms, and Examples, 2022. Disponível em: https://www.spiceworks.com/tech/devops/articles/what-is-dynamic-programming/. Acesso em: 25 de jun. de 2023.