Programação Dinamica

DP

Programação dinâmica é uma técnica para resolver problemas complexos dividindo-os em subproblemas mais simples e resolvendo cada subproblema uma única vez.

Essa técnica pode ser vista como uma variação do trade-off espaço-por-tempo.

Exemplos:

- Sequência de Fibonacci: Calcula os números de Fibonacci armazenando resultados anteriores para evitar cálculos repetidos.
- Caminho Mais Curto em um Grafo:
 Algoritmos como <u>Dijkstra ou Bellman-Ford</u>
 encontram o caminho mais curto de um ponto a todos os outros em um grafo.

Abordagens para Implementar Programação Dinâmica

Bottom-up (De Baixo para Cima):

- Começa resolvendo os problemas menores primeiro e gradualmente constrói soluções para problemas maiores.
- Implementação: Inicia com casos base conhecidos e avança iterativamente para resolver problemas maiores.
- Vantagens: Não precisa armazenar resultados intermediários desnecessários, o que economiza espaço.
- Exemplo: Útil para problemas como o da fila de moedas, onde cada etapa depende apenas das soluções das etapas anteriores.

Top-down (De Cima para Baixo):

- Começa com o problema maior e o divide em subproblemas menores.
- Técnica: Utiliza memoização, armazenando resultados de subproblemas para evitar recálculos.
- Vantagens: Pode economizar tempo ao calcular apenas as partes necessárias da solução completa.
- Exemplo: Útil quando é possível evitar calcular todas as soluções possíveis, focando apenas nas partes necessárias, como no problema de troco.

Algoritmos de aproximação

Às vezes é ainda mais difícil encontrar uma solução para certos problemas.
Alternativa: "resolver" o problema usando algoritmos de aproximação. A solução não precisa ser ótima, apenas aceitável. Exemplos: algoritmos de aproximação gananciosos para o Problema do Caixeiro Viajante (TSP).

- Algoritmo do vizinho mais próximo.
- Algoritmo baseado em MST (Árvore Geradora Mínima).

TSP: Algoritmo do Vizinho Mais Próximo Descrição:

- Nodo inicial: a.
- Caminho: a b c d a.
- Custo: 10.

Solução Ótima:

- Caminho: a b d c a.
- Custo: 8.

Taxa de Precisão:

• r(sa)=10/8=1,25.

Alterando o peso de (a, d) para w:

• r(sa)=(4+w)/8.

Problema da Fila de Moedas (Coin-Row Problem)

- Neste problema, temos uma fila de moedas com valores c1,c2,...,cn. O objetivo é escolher um conjunto de moedas de modo a maximizar o valor total, sem pegar duas moedas adjacentes.
- A recursão para resolver esse problema é dada por:
 F(n)=max{cn+F(n-2),F(n-1)}, para n > 1
- ∘ F(n) é o máximo valor que pode ser obtido até o n-ésimo elemento da fila.
- ∘ F(0)=0 (valor máximo com 0 moedas).
- ∘ F(1)=c1 (valor máximo com apenas a primeira moeda).

Problema da Mochila 0-1

- Temos n itens, cada um com um peso w[i] e um valor v[i].
- Temos uma capacidade máxima W para a mochila.
- O objetivo é maximizar o valor total dos itens colocados na mochila sem exceder a capacidade.
- 1. Inicializamos uma matriz F de dimensões (n+1)×(W+1).
- 2. Para cada item i (de 0 até n) e cada capacidade i (de 0 até W):
 - Se i ou j for 0, definimos F[i][j] como 0 (caso base: mochila vazia ou capacidade zero).
 - Se o peso do item atual w[i] for menor ou igual à capacidade atual j:
 - Calculamos o valor máximo entre não incluir o item atual F[i-1][j] e incluir o item atual v[i]+F[i-1][j-w[i]].
 - Se o peso do item atual w[i] for maior que a capacidade atual j, não podemos incluir o item, então F[i][j] é igual ao valor sem o item atual F[i-1][j]
- 3. Retornamos F[n][W], que contém o valor máximo para n itens e capacidade W.

Passos Gerais:

- 1. Construa uma Árvore Geradora Mínima (MST).
- 2. Realize um passeio ao redor da MST (registrando a ordem dos nós visitados).
- 3. Elimine todas as ocorrências repetidas, exceto o nó inicial (fazendo atalhos no passeio).

Explicação

Para resolver problemas difíceis como o TSP, onde encontrar a solução ótima é complexo e demorado, usamos algoritmos de aproximação que fornecem soluções boas o suficiente, de maneira mais rápida.

Algoritmo do Vizinho Mais Próximo:

- Começa em um nó inicial.
- Sempre vai para o nó mais próximo ainda não visitado.
- Pode não encontrar a solução ótima, mas é simples e rápido.

Algoritmo Baseado em MST:

- Constrói uma Árvore Geradora Mínima, que conecta todos os nós com o menor custo total possível.
- Faz um passeio ao redor da MST, visitando cada nó e registrando a ordem.
- Remove as repetições, mantendo apenas o nó inicial para formar um ciclo.