DCC206 – Algoritmos 1

Aula 17 – Programação Dinâmica – Parte 4

Professor Renato Vimieiro

DCC/ICEx/UFMG

Introdução

- Na aula de hoje veremos a aplicação de programação de dinâmica em um problema com bastante apelo prático: o problema da mochila
- Anteriormente nós exploramos esse problema no contexto de algoritmos gulosos
- A diferença entre as duas abordagens é que a última trabalhava sobre uma versão em que poderíamos pegar frações de um item
- Agora vamos nos concentrar na versão em que os itens só podem ser pegos integralmente ou deixados para trás

Problema da mochila binário

- Considere que um estudante vai sair em viagem e precisa fazer suas malas
- Ele tem n itens para transportar
- Cada item i tem um valor v_i associado (e.g. a importância daquele item para o estudante na viagem). Porém, cada item tem um peso w_i associado
- Como a companhia aérea impõe restrições no limite de bagagem (no máximo W quilos podem ser transportados), o estudante precisa escolher quais itens serão levados e quais serão deixados para trás
- Naturalmente, ele quer obter o maior valor total
- A pergunta é como resolver esse problema

Problema da mochila binário

- Uma abordagem ingênua para resolver o problema consiste em explorar todos os possíveis subconjuntos de itens
- Obviamente, essa abordagem é inviável para grandes quantidades de itens, já que ela tem custo exponencial
- Assim, ele é um forte candidato ao desenvolvimento de uma abordagem baseada em programação dinâmica

Problema da mochila binário

- Como sempre, o primeiro passo é verificar que o problema apresenta subestrutura ótima
- Vamos definir que o valor ótimo para transportar os itens seja dado pela função M(n,W)
- Se analisarmos o último item, ele pode ou não fazer parte da solução
- Caso ele faça parte da solução, então o valor ótimo $M(n,W) = v_n + M(n-1,W-w_n)$
    - Seria o valor de n mais o valor ótimo alcançável com n-1 itens e capacidade W menos o peso de n
- Se n não fizer parte da solução ótima, então $M(n,W) = M(n-1,W)$
    - Simplesmente descartamos o item n e verificamos uma solução ótima sem ele
- Note que, se $w_i > W'$, ou seja, um item tiver capacidade maior que a disponível no momento da sua escolha, então ele obrigatoriamente não pode ser escolhido. Logo somente a segunda opção é válida. Portanto, a solução ótima de um problema depende das soluções ótimas dos subproblemas.

Problema da mochila binário

- O segundo passo é desenvolver uma relação de recorrência para resolver o problema
- A dedução feita para demonstrar a subestrutura ótima atende parcialmente a esse propósito
- A única observação que temos é que, se um item puder ser escolhido, então temos a opção de colocá-lo ou não na solução. Devemos escolher a melhor dentre elas.
- $\mathrm{M(i,j)} = \begin{cases} \mathrm{M(i-1,j)} & \text{se } w_i > \mathrm{j} \\ \max\{\mathrm{M(i-1,j)}, \mathrm{M(i-1, j-} w_i) + v_i\} & \text{se } w_i \leq \mathrm{j} \end{cases}$
- Além disso, sabemos que M(0,j) = M(i,0) = 0

Problema da mochila binário

- O terceiro passo é desenvolver a solução iterativa bottom-up a partir da relação de recorrência
- Essencialmente, cada valor de M depende somente de subproblemas menores (na mesma coluna e linha superior, ou linha superior e algumas colunas anteriores)
- Assim, podemos preencher a matriz linha a linha

Problema da mochila binário

- M = matriz (n+1) x (W+1)
- Para i = 0 até n: M[i][0] = 0
- Para j = 0 até W: M[0][j] = 0
- Para k = 1 até n:
  - Para X=1 até W:
    - Se w_k > X, então M[k][X] = M[k-1][X]
    - Senão M[k][X] = max(M[k-1][X], v_k + M[k-1][X-w_k])
- Retorne M

Problema da mochila binário

- Exemplo: W=5

Item	v_i	w_i
1	12	2
2	10	1
3	20	3
4	15	2

Problema da mochila binário

- O custo desse algoritmo é o de preencher a matriz O(nW)
- Esse tipo de algoritmo é conhecido como pseudo-polinomial
- Note que ele é polinomial na magnitude de W e não no tamanho da sua representação
- Ou seja, podemos fazer W se comportar como uma função exponencial do tamanho da entrada
- Assim, o custo total do algoritmo será exponencial em função do tamanho da entrada
  - Ele depende da magnitude de W e não no tamanho da sua representação
- É possível reconstruir os itens escolhidos a partir do valor da solução final
  - Esse quarto passo ficará como exercício

Leitura

- Seção 8.2 (Levitin)
- Capítulo 6.4 (Kleinberg e Tardos)
- Seção 12.6 (Goodrich e Tamassia)