Projeto e Análise de Algoritmos

Programação Dinâmica

Introdução

- Inicia-se com a menor e mais simples sub-instância
- Abordagem bottom-up
- Combinaçao de respostas das sub-instâncias de tamanho crescente, obtem-se a resposta desejada

Introdução

- Aplicável quando os subproblemas não são independentes, isto é, quando os subproblemas compartilham subsubproblemas.

- Um algoritmo de programação dinâmica resolve cada subproblema uma vez só e então grava sua resposta em uma tabela

Exemplo

- Cálculo dos coeficientes binomiais

$$\binom{n}{k} = \begin{cases} 1 & \text{se } k = 0 \text{ ou } k = n \\ \binom{n-1}{k-1} + \binom{n-1}{k} & \text{se } 0 < k < n \\ 0 & \text{caso contrário} \end{cases}$$

```
Funcao C(n,k)
    if k=0 ou k=n return 1
    else return C(n-1,k-1)+C(n-1,k)
```

Como resolver de forma eficiente???

Exemplo

- Cálculo dos coeficientes binomiais

k

n

1								
1	1							
1	2	1						
1	3	3	1					
1	4	6	4	1				
1	5	10	10	5	1			
1	6	15	20	15	6	1		
		C						

Exemplo: troco ou pagamento

- Qual o menor número de moedas que devem ser usadas para pagar uma conta, uma vez que estao disponíveis moedas de
  - 1 real (a)
  - 25 centavos (b)
  - 10 centavos (c)
  - 5 centavos (d)
  - 1 centavo (e)
- Instância do problema: 2,89 reais
  - 10 moedas (2a + 3b + 1c + 4e)

Exemplo: troco ou pagamento

- Algoritmo guloso para resolver este problema
  - Inicie com nada
  - Cada estágio, sempre adicione a moeda de maior valor possível que não ultrapasse o valor tota
- Programaçao dinâmica
  - Ponto fundamental é configurar os resultados intermediários para que sejam combinados de forma a gerar o resultado desejado

Exemplo

- Considere o seguinte conjunto de moedas
  - 1 centavo
  - 4 centavos
  - 6 centavos

- Quantas moedas são necessárias para compor 8 centavos?

Algoritmo de PD

Soma	0	1	2	3	4	5	6	7	8
D_1 =1	0	1	2	3	4	5	6	7	8
D_2 =4	0	1	2	3	1	2	3	4	2
D_3 =6	0	1	2	3	1	2	1	2	2

- $C[I,J] = MIN(C[I-1,J],1+C[I,J-D_i])$

Problema da mochila

- O objetivo é encher a mochila de forma a maximizar o número de objetos incluídos sem exceder o limite de peso.
- Os objetos NÃO podem ser divididos em pedaços menores

- Como implementar?
- Custo computacional?

Algoritmo de PD

PESO	0	1	2	3	4	5	6	7	8	9	10	11
$W_1=1,\ v_1=1$	0	1	1	1	1	1	1	1	1	1	1	1
$W_2=2,\ v_2=6$	0	1	6	7	7	7	7	7	7	7	7	7
$W_3=5,\ v_3=18$	0	1	6	7	7	18	19	24	25	25	25	25
$W_4=6,\ v_4=22$	0	1	6	7	7	18	22	24	28	29	29	40
$W_5=7,\ v_5=28$	0	1	6	7	7	18	22	28	29	34	35	40

- $C[I,J] = MAX(C[I-1,J], v_i + C[I-1,J-w_i])$

Exemplo - Fibonacci

1. Use PD para encontrar F(50) da série de Fibonacci

Como resolver ?????

Exemplo – Subsequência comum

1. Subsequência comum mais longa
   1. G A A T T C A G T T A
   2. G G A T C G A

   Como resolver????

Exemplo – Subsequência comum

		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0	0	0	0	0
G	0	1	1	1	1	1	1	1	1	1	1	1
G	0	1	1	1	1	1	1	1	2	2	2	2
A	0	1	2	2	2	2	2	2	2	2	2	3
T	0	1	2	2	3	3	3	3	3	3	3	3
C	0	1	2	2	3	3	4	4	4	4	4	4
G	0	1	2	2	3	3	4	4	5	5	5	5
A	0	1	3	3	3	3	4	5	5	5	5	6

Custo de transformaçao entre duas sequências

- 1 3 4 3 3 4 5 4 5 6 7 8 3 3 3 4 5
- 1 3 4 3 3 3 3 4 5 4 5 6 7 8 3 3 3

Como resolver????????

Menores Caminhos

- Considere um grafo direcionado G = (V,E), onde ***n*** é o número de vértices e ***e*** o número de arestas.
- Cada aresta tem um comprimento não negativo

PROBLEMA

Determinar o comprimento de menor caminho, entre quaisquer pares de vértices

- Como implementar?
- Custo computacional?

Menores Caminhos

Matriz de adjacências de um Grafo G

0	5	∞	∞
50	0	15	5
30	∞	0	15
15	∞	5	0

Menores Caminhos

0	5	∞	∞
50	0	15	5
30	∞	0	15
15	∞	5	0

0	5	∞	∞
50	0	15	5
30	35	0	15
15	20	5	0

0	5	20	10
50	0	15	5
30	35	0	15
15	20	5	0

0	5	20	10
40	0	15	5
30	35	0	15
15	20	5	0

D[I,J]=min(d[I,J],D[I,K]+D[K,J])

0	5	15	10
20	0	10	5
30	35	0	15
15	20	5	0

Reconhecimento aproximado

- Algoritmos buscam reconhecer padrões parecidos, considerando:
  - Erros ortográficos (ex.: atensão)
  - Mesmos fonemas (ex.: paço e passo)
  - Variações ortográficas (ex.: Luiz e Luís)
  - Etc.

Reconhecimento aproximado

- Aplicações:
  - Corretores ortográficos
  - Comparação de DNA
  - Filtragem de SPAM
  - OCR (optical character recognition)

Reconhecimento aproximado

- Distância de edição
  - Mede a diferença entre duas sequências
  - A diferença é dada pelo número de edições (inserções, exclusões e substituições) necessárias para transformar uma sequência em outra.
- Exemplos:
  - Gato → Pato (1 edição)
  - Paço → Passo (2 edições)

Reconhecimento aproximado

- Distância de Levenshtein (1965)

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & se \min(i,j) = 0 \\ \min \begin{cases} lev_{a,b}(i-1,j) & +1 \\ lev_{a,b}(i,j-1) & +1 \\ lev_{a,b}(i-1,j-1) + C \end{cases} & nos\ outros\ casos \end{cases}$$

em que $C = 0$, se $a_i = b_j$, ou $C = 1$, se $a_i \neq b_j$.

Reconhecimento aproximado

- Exemplo 1:
  CASACO x CASCAO

		C	A	S	A	C	O
	0	1	2	3	4	5	6
C	1						
A	2						
S	3						
C	4						
A	5						
O	6						

Reconhecimento aproximado

- Exemplo 1:
  CASACO x CASCAO

		C	A	S	A	C	O
	0	1	2	3	4	5	6
C	1	0	1	2	3	4	5
A	2	1	0	1	2	3	4
S	3	2	1	0	1	2	3
C	4	3	2	1	1	1	2
A	5	4	3	2	2	2	2
O	6	5	4	3	3	3	2

Reconhecimento aproximado

- Exemplo 1:
  PIGARRO x CIGANO

		P	I	G	A	R	R	O
	0	1	2	3	4	5	6	7
C	1	1	2	3	4	5	6	7
I	2	2	1	2	3	4	5	6
G	3	3	2	1	2	3	4	5
A	4	4	3	2	1	2	3	4
N	5	5	4	3	2	2	3	4
O	6	6	5	4	3	3	3	3