

# Aula II - Distância de Edição

Luís Felipe

UFF

02 de Outubro de 2023

# Programação Dinâmica

- Na aula passada vimos como podemos aproveitar **soluções de subproblemas** para obter a solução do problema.
- Para tal, construímos a solução da menor para a maior instância, e em cada passo, tomamos a **melhor subsolução** possível para obter a atual.
- Vimos como a PD pode nos ajudar a eventualmente resolver um problema **eficientemente**, ao passo que pela divisão e conquista pode eventualmente ser **ineficiente**.

**Exemplos?**

- Vamos aplicar essa técnica para resolver um novo problema: **Distância de Edição**.

Luis Felipe  
02/10/23

## Distância de Edição - Levenshtein

- Quando escrevemos algo errado, qual critério de **proximidade** de alguma outra palavra um corretor adota?
- Uma medida de distância entre duas strings é dada pela forma em que elas são **alinhas**, ou combinadas.
- Um **alinhamento** é uma forma escrever duas strings uma abaixo da outra.

Ex1

S	-	N	O	W	Y
S	U	N	N	-	Y

Ex2

-	S	N	O	W	-	Y
S	U	N	-	-	N	Y

- Qual **melhor alinhamento**?! Aquele que tem maior número de "casamentos" de letras iguais. Ou seja, aquele que tem o menor número de casamentos de **letras distintas** ou **espaços vazios** (gap, símbolo -).

**Obs.:** Um gap pode ter custo maior (ou menor) do que um casamento de letras distintas, depende do contexto.

Luis Felipe

02/10/23

## Distância de Edição

- O **custo** de um alinhamento é o número de colunas que as letras diferem.
- Ex 1: custo 3; Ex 2: custo 5.
- A **distância de edição** entre duas strings é o custo do melhor alinhamento possível.
- Uma edição é justamente uma operação de **inserção**, **deleção** e **substituição** feita ao alinhar uma sequência com outra.

**Obs.:** Buscando uma solução por força bruta, seria eficiente?

Luís Felipe  
02/10/23

# Programação Dinâmica

- Para ter uma PD, devemos conseguir encontrar a propriedade chave (visto na **Aula 10**):
  - ▶ Existe uma ordem do problema em subproblemas e uma relação que mostra como resolver um subproblema baseado nas respostas de subproblemas **menores**, ou seja, subproblemas que aparecem antes na ordem.
- Para encontrar a distância de edição de duas strings  $x[1 \dots m]$  e  $y[1 \dots n]$  quais são bons subproblemas?
- Para isso, vamos analisar a distância de edição de cada prefixo da primeira string,  $x[1 \dots i]$ , com todos os prefixos da segunda string,  $y[1 \dots j]$ . Esse subproblema é denotado por  $E(i, j)$ .
- Ao final, nosso objetivo é computar  $E(m, n)$ .
- $E(7, 5)$ :

E X P O N E N T I A L  
P O L Y N O M I A L

Luís Felipe  
02/10/23

## Programação Dinâmica - Analisando prefixos

- Para montar a PD precisamos expressar  $E(i, j)$  em termos dos subproblemas menores.
- Em relação aos alinhamentos entre  $x[1 \dots i]$  e  $y[1 \dots j]$ , a **coluna mais a direita** do melhor alinhamento só pode ser de uma das três formas:

$$\begin{array}{ccc} 1. & x[i] & 2. & - & 3. & x[i] \\ & - & & y[j] & & y[j] \end{array}$$

1. Esse alinhamento **adiciona custo 1** (uma vez que houve um gap), e resta agora alinhar  $x[1 \dots i - 1]$  com  $y[1 \dots j]$ . Ou seja, recaímos no problema  $E(i - 1, j)$ .
2. Esse alinhamento **adiciona custo 1** (uma vez que houve um gap), e resta agora alinhar  $x[1 \dots i]$  com  $y[1 \dots j - 1]$ . Ou seja, recaímos no problema  $E(i, j - 1)$ .
3. Esse alinhamento **adiciona custo 1** (caso  $x[i] \neq y[j]$ ) **ou 0** (caso  $x[i] = y[j]$ ), e resta agora alinhar  $x[1 \dots i - 1]$  com  $y[1 \dots j - 1]$ . Ou seja, recaímos no problema  $E(i - 1, j - 1)$ .

Luis Felipe  
02/10/23

## Programação Dinâmica - Relação dos subproblemas

- Expressamos  $E(i, j)$  em função dos três subproblemas  $E(i-1, j)$ ,  $E(i, j-1)$  e  $E(i-1, j-1)$ . Tomamos dentre esses três, aquele que tem o melhor custo (**menor custo**):

$$E(i, j) = \min\{1 + E(i-1, j), 1 + E(i, j-1), \text{diff}(i, j) + E(i-1, j-1)\},$$

onde  $\text{diff}(i, j) = 0$ , se  $x[i] = y[j]$  ou  $\text{diff}(i, j) = 1$ , se  $x[i] \neq y[j]$ .

- **Ex:**  $E(4, 3)$  corresponde a **EXPO** e **POL**. A coluna mais a direita será:  $E(4, 3) = \min\{1 + E(3, 3), 1 + E(4, 2), 1 + E(3, 2)\}$ .

0	-	0
-	L	L

- Só falta sabermos os **casos base** das chamadas aos subproblemas. Isso ocorre quando  $E(0, j)$  ou  $E(i, 0)$ .
  - ▶  $E(0, j)$  é quando alinhamos a 1ª string usando  $j$  gaps e a 2ª string usando os  $j$  primeiros elementos.  $E(0, j) = j$ .
  - ▶  $E(i, 0)$  é quando alinhamos a 1ª string usando os  $i$  primeiros elementos e a 2ª string usando  $i$  gaps.  $E(i, 0) = i$ .

Luis Felipe  
02/10/23

## Programação Dinâmica - Em sumário

- Note que queremos computar cada  $E(i, j)$ , com  $i = 0, \dots, m$  e  $j = 0, \dots, n$ . Assim, temos  $(m + 1) \cdot (n + 1)$  células para preencher os valores de todos os  $E(i, j)$ .
- Inicialmente, começamos preenchendo  $E(i, 0)$  e depois  $E(0, j)$ . Após isso, conseguimos obter os valores de todos os demais consultando os subproblemas menores.

Para  $i = 0, \dots, m$ :

$$E(i, 0) = i$$

Para  $j = 1, \dots, n$ :

$$E(0, j) = j$$

Para  $i = 1, \dots, m$ :

Para  $j = 1, \dots, n$ :

$$E(i, j) = \min\{1 + E(i - 1, j), 1 + E(i, j - 1), \text{diff}(i, j) + E(i - 1, j - 1)\}$$

Retorne  $E(m, n)$

- Cada etapa consultamos 3 subproblemas na tabela e fazemos uma soma, assim a complexidade é  $O(m \cdot n)$ .



Luís Felipe

02/10/23

## Programação Dinâmica - DAG

- **DAG**: Cada subproblema é um vértice e cada aresta direcionada é de um subproblema predecessor a um sucessor.
- Assim, na posição  $(i, j)$  há as arestas vindo de  $(i - 1, j)$ , de  $(i, j - 1)$  e de  $(i - 1, j - 1)$ .
- Os caminhos pelas arestas associadas a vértices de menores custos partindo de  $(m, n)$  até  $(0, 0)$  dão o alinhamento ótimo.



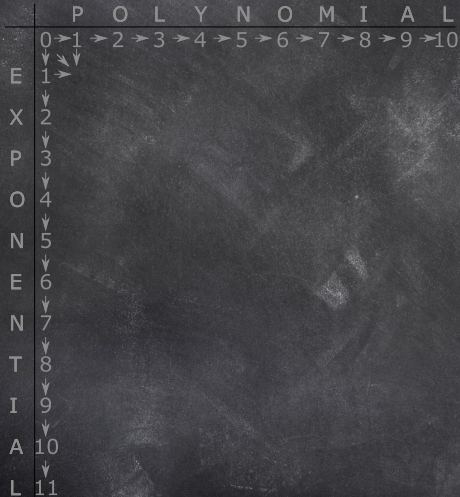






Luis Felipe  
02/10/23

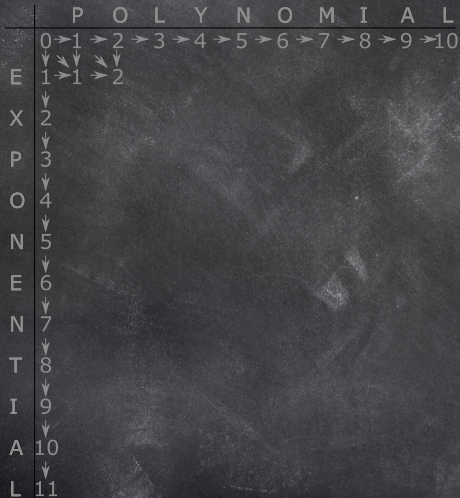
# Programação Dinâmica - Execução





Luis Felipe  
02/10/23

# Programação Dinâmica - Execução







Luis Felipe  
02/10/23

# Programação Dinâmica - Execução

	P	O	L	Y	N	O	M	I	A	L	
	0	1	2	3	4	5	6	7	8	9	10
E	1	1	2	3	4	5	6	7	8	9	10
X	2	2	2	3	4	5	6	7	8	9	10
P	3	2	3	3	4	5	6	7	8	9	10
O	4	3	2	3	4	5	5	6	7	8	9
N	5	4	3	3	4	4	5	6	7	8	9
E	6	5	4	4	4	5	5	6	7	8	9
N	7	6	5	5	5	4	5	6	7	8	9
T	8	7	6	6	6	5	5	6	7	8	9
I	9	8	7	7	7	6	6	6	6	7	8
A	10	9	8	8	8	7	7	7	7	6	7
L	11	10	9	8	9	8	8	8	8	7	6

Luis Felipe  
02/10/23

# Programação Dinâmica - Execução

	P	O	L	Y	N	O	M	I	A	L	
	0	1	2	3	4	5	6	7	8	9	10
E	1	1	2	3	4	5	6	7	8	9	10
X	2	2	2	3	4	5	6	7	8	9	10
P	3	2	3	3	4	5	6	7	8	9	10
O	4	3	2	3	4	5	5	6	7	8	9
N	5	4	3	3	4	4	5	6	7	8	9
E	6	5	4	4	4	5	5	6	7	8	9
N	7	6	5	5	5	4	5	6	7	8	9
T	8	7	6	6	6	5	5	6	7	8	9
I	9	8	7	7	7	6	6	6	6	7	8
A	10	9	8	8	8	7	7	7	7	6	7
L	11	10	9	8	9	8	8	8	8	7	6

Luis Felipe  
02/10/23

# Programação Dinâmica - Execução

	P	O	L	Y	N	O	M	I	A	L	
	0	1	2	3	4	5	6	7	8	9	10
E	1	1	2	3	4	5	6	7	8	9	10
X	2	2	2	3	4	5	6	7	8	9	10
P	3	2	3	3	4	5	6	7	8	9	10
O	4	3	2	3	4	5	5	6	7	8	9
N	5	4	3	3	4	4	5	6	7	8	9
E	6	5	4	4	4	5	5	6	7	8	9
N	7	6	5	5	5	4	5	6	7	8	9
T	8	7	6	6	6	5	5	6	7	8	9
I	9	8	7	7	7	6	6	6	6	7	8
A	10	9	8	8	8	7	7	7	7	6	7
L	11	10	9	8	9	8	8	8	8	7	6

Luis Felipe  
02/10/23

# Programação Dinâmica - Execução

	P	O	L	Y	N	O	M	I	A	L	
	0	1	2	3	4	5	6	7	8	9	10
E	1	1	2	3	4	5	6	7	8	9	10
X	2	2	2	3	4	5	6	7	8	9	10
P	3	2	3	3	4	5	6	7	8	9	10
O	4	3	2	3	4	5	5	6	7	8	9
N	5	4	3	3	4	4	5	6	7	8	9
E	6	5	4	4	4	5	5	6	7	8	9
N	7	6	5	5	5	4	5	6	7	8	9
T	8	7	6	6	6	5	5	6	7	8	9
I	9	8	7	7	7	6	6	6	6	7	8
A	10	9	8	8	8	7	7	7	7	6	7
L	11	10	9	8	9	8	8	8	8	7	6

L  
L

Luis Felipe  
02/10/23

# Programação Dinâmica - Execução

	P	O	L	Y	N	O	M	I	A	L	
	0	1	2	3	4	5	6	7	8	9	10
E	1	1	2	3	4	5	6	7	8	9	10
X	2	2	2	3	4	5	6	7	8	9	10
P	3	2	3	3	4	5	6	7	8	9	10
O	4	3	2	3	4	5	5	6	7	8	9
N	5	4	3	3	4	4	5	6	7	8	9
E	6	5	4	4	4	5	5	6	7	8	9
N	7	6	5	5	5	4	5	6	7	8	9
T	8	7	6	6	6	5	5	6	7	8	9
I	9	8	7	7	7	6	6	6	6	7	8
A	10	9	8	8	8	7	7	7	7	6	7
L	11	10	9	8	9	8	8	8	8	7	6

- T I A L  
O M I A L

Luis Felipe  
02/10/23

# Programação Dinâmica - Execução

	P	O	L	Y	N	O	M	I	A	L		
	0	1	2	3	4	5	6	7	8	9	10	
E	1	1	2	3	4	5	6	7	8	9	10	
X	2	2	2	3	4	5	6	7	8	9	10	
P	3	2	3	3	4	5	6	7	8	9	10	
O	4	3	2	3	4	5	5	6	7	8	9	
N	5	4	3	3	4	4	5	6	7	8	9	
E	6	5	4	4	4	5	5	6	7	8	9	
N	7	6	5	5	5	5	4	5	6	7	8	9
T	8	7	6	6	6	6	5	5	6	7	8	9
I	9	8	7	7	7	7	6	6	6	6	7	8
A	10	9	8	8	8	8	7	7	7	7	6	7
L	11	10	9	8	8	8	8	8	8	7	6	6

N E N - T I A L  
L Y N O M I A L

Luis Felipe  
02/10/23

# Programação Dinâmica - Execução

	P	O	L	Y	N	O	M	I	A	L		
	0	1	2	3	4	5	6	7	8	9	10	
E	1	1	2	3	4	5	6	7	8	9	10	
X	2	2	2	3	4	5	6	7	8	9	10	
P	3	2	3	3	4	5	6	7	8	9	10	
O	4	3	2	3	4	5	5	6	7	8	9	
N	5	4	3	3	4	4	5	6	7	8	9	
E	6	5	4	4	4	5	5	6	7	8	9	
N	7	6	5	5	5	5	4	5	6	7	8	9
T	8	7	6	6	6	6	5	5	6	7	8	9
I	9	8	7	7	7	7	6	6	6	6	7	8
A	10	9	8	8	8	8	7	7	7	7	6	7
L	11	10	9	8	8	8	8	8	8	7	6	6

O N E N - T I A L  
O L Y N O M I A L



Luis Felipe  
02/10/23

# Programação Dinâmica - Execução

	P	O	L	Y	N	O	M	I	A	L		
	0	1	2	3	4	5	6	7	8	9	10	
E	1	1	2	3	4	5	6	7	8	9	10	
X	2	2	2	3	4	5	6	7	8	9	10	
P	3	2	3	3	4	5	6	7	8	9	10	
O	4	3	2	3	4	5	5	6	7	8	9	
N	5	4	3	3	4	4	5	6	7	8	9	
E	6	5	4	4	4	5	5	6	7	8	9	
N	7	6	5	5	5	5	4	5	6	7	8	9
T	8	7	6	6	6	6	5	5	6	7	8	9
I	9	8	7	7	7	7	6	6	6	6	7	8
A	10	9	8	8	8	8	7	7	7	7	6	7
L	11	10	9	8	9	8	8	8	8	8	7	6

P O N E N - T I A L  
P O L Y N O M I A L

Luis Felipe  
02/10/23

# Programação Dinâmica - Execução

	P	O	L	Y	N	O	M	I	A	L		
	0	1	2	3	4	5	6	7	8	9	10	
E	1	1	2	3	4	5	6	7	8	9	10	
X	2	2	2	3	4	5	6	7	8	9	10	
P	3	2	3	3	4	5	6	7	8	9	10	
O	4	3	2	3	4	5	5	6	7	8	9	
N	5	4	3	3	4	4	5	6	7	8	9	
E	6	5	4	4	4	5	5	6	7	8	9	
N	7	6	5	5	5	5	4	5	6	7	8	9
T	8	7	6	6	6	6	5	5	6	7	8	9
I	9	8	7	7	7	7	6	6	6	6	7	8
A	10	9	8	8	8	8	7	7	7	7	6	7
L	11	10	9	8	9	8	8	8	8	8	7	6

X P O N E N - T I A L  
- P O L Y N O M I A L

Luis Felipe  
02/10/23

# Programação Dinâmica - Execução

	P	O	L	Y	N	O	M	I	A	L		
	0	1	2	3	4	5	6	7	8	9	10	
E	1	1	2	3	4	5	6	7	8	9	10	
X	2	2	2	3	4	5	6	7	8	9	10	
P	3	2	3	3	4	5	6	7	8	9	10	
O	4	3	2	3	4	5	5	6	7	8	9	
N	5	4	3	3	4	4	5	6	7	8	9	
E	6	5	4	4	4	5	5	6	7	8	9	
N	7	6	5	5	5	5	4	5	6	7	8	9
T	8	7	6	6	6	6	5	5	6	7	8	9
I	9	8	7	7	7	7	6	6	6	6	7	8
A	10	9	8	8	8	8	7	7	7	7	6	7
L	11	10	9	8	9	8	8	8	8	8	7	6

E X P O N E N T I A L  
- - P O L Y N O M I A L

Luis Felipe  
02/10/23

# Programação Dinâmica - Execução

	P	O	L	Y	N	O	M	I	A	L		
	0	1	2	3	4	5	6	7	8	9	10	
E	1	1	2	3	4	5	6	7	8	9	10	
X	2	2	2	3	4	5	6	7	8	9	10	
P	3	2	3	3	4	5	6	7	8	9	10	
O	4	3	2	3	4	5	5	6	7	8	9	
N	5	4	3	3	4	4	5	6	7	8	9	
E	6	5	4	4	4	5	5	6	7	8	9	
N	7	6	5	5	5	5	4	5	6	7	8	9
T	8	7	6	6	6	6	5	5	6	7	8	9
I	9	8	7	7	7	7	6	6	6	6	7	8
A	10	9	8	8	8	8	7	7	7	7	6	7
L	11	10	9	8	9	8	8	8	8	8	7	6

E X P O N E N T I A L  
- - P O L Y N O M I A L