Prof. Eurinardo

Aula Passada

Linha de Montagem

Evenno

Algorita

Recursivo

Memoização

Aula 16 Programação Dinâmica Linha de Montagem

Projeto e Análise de Algoritmos

Professor Eurinardo Rodrigues Costa Universidade Federal do Ceará Campus Russas

2021.1

Linha de Montagem

Problema

Exemplo

Algoritmos

Recursivo

Memoização

inha de

Problema

Exempl

Algoritmos

necursivo

Linha de Montagem

Exemplo

Algoritmo

ecursivo

Programação Dinâmica

Definição (Programação Dinâmica)

Linha de Montagem

Exemplo

Algoritmo

Recursivo

Programação Dinâmica

Definição (Programação Dinâmica)

Seguir os passos:

(1) Verificar propriedade de subestrutura ótima

Definição (Programação Dinâmica)

- (1) Verificar propriedade de subestrutura ótima
- (2) Obter uma recursão para o valor ótimo do problema

Definição (Programação Dinâmica)

- (1) Verificar propriedade de subestrutura ótima
- (2) Obter uma recursão para o valor ótimo do problema
- (3) Obter um algoritmo Bottom-UP para calcular o valor ótimo

Memoização

Definição (Programação Dinâmica)

- (1) Verificar propriedade de subestrutura ótima
- (2) Obter uma recursão para o valor ótimo do problema
- (3) Obter um algoritmo Bottom-UP para calcular o valor ótimo
- (4) Obter uma solução ótima do problema.

Linha de Montagem

PAA - Aula 16

Prof. Eurinardo

Aula Passada

Linha de

Problema

Exemplo

Algoritmo

Recursivo

PAA - Aula 16

Prof. Eurinardo

Aula Passada

Linha de

Montagem Problema

Exemplo

Algoritmos

Memoização

Algoritmo 2: Linha-Rec(a, t, e, x, n)

Entrada: $a, t \rightarrow \text{matrizes } 2 \times n$

PAA - Aula 16

Prof. Eurinardo

Aula Passada

Linha de Montagem

Problema

Exemplo

Algoritme

Recursivo

Memoização



Algoritmo 3: Linha-Rec(a, t, e, x, n)

Entrada: $a, t \rightarrow \text{matrizes } 2 \times n$

 $e, x \rightarrow$ vetores de tamanho 2

PAA - Aula 16

Prof. Eurinardo

Aula Passada

Linha de Montagem

Problema

Exemplo

December

Recursivo

Memoização
Programação Dinâmica

Algoritmo 4: Linha-Rec(a, t, e, x, n)

Entrada: $a, t \rightarrow \text{matrizes } 2 \times n$

 $e, x \rightarrow$ vetores de tamanho 2

 $n \rightarrow$ número de estações

PAA - Aula 16

Prof. Eurinardo

Aula Passada

Linha de Montagem

Problema

Exemplo

Decimina

Recursivo

Memoização Programação Dinâmica

Aula Passada

Linha de Montagem

PAA - Aula 16

Prof. Eurinardo

Memoização

Algoritmo 5: Linha-Rec(a, t, e, x, n)

Entrada: $a, t \rightarrow \text{matrizes } 2 \times n$

 $e, x \rightarrow \text{vetores de tamanho 2}$

 $n \rightarrow$ número de estações

Saída: Valor Ótimo

Recursivo

Memoização

Algoritmo 6: Linha-Rec(a, t, e, x, n)

Entrada: $a, t \rightarrow \text{matrizes 2} \times n$

 $e, x \rightarrow$ vetores de tamanho 2

 $n \rightarrow$ número de estações

Saída: Valor Ótimo

1 **criar vetor** $f_1[1 \cdots n]$, $f_2[1 \cdots n]$ com -1's

Algoritmos

Recursivo

Memoização Programação Dinâmica

Algoritmo 7: Linha-Rec(a, t, e, x, n)

Entrada: $a, t \rightarrow \text{matrizes } 2 \times n$

 $e, x \rightarrow$ vetores de tamanho 2

 $n \rightarrow$ número de estações

Saída: Valor Ótimo

1 criar vetor $f_1[1 \cdots n]$, $f_2[1 \cdots n]$ com -1's

2
$$f_1[1] \leftarrow e_1 + a_{1,1}$$

Algoritmo 8: Linha-Rec(a, t, e, x, n)

Entrada: $a, t \rightarrow \text{matrizes } 2 \times n$

 $e, x \rightarrow$ vetores de tamanho 2

 $n \rightarrow$ número de estações

Saída: Valor Ótimo

1 criar vetor $f_1[1 \cdots n]$, $f_2[1 \cdots n]$ com -1's

2
$$f_1[1] \leftarrow e_1 + a_{1,1}$$

3
$$f_2[1] \leftarrow e_2 + a_{2,1}$$

Prof. Eurinardo

Aula Passada

Linha de Montagem

Evennele

Exemplo

Becursivo

Memoização

Algoritmo 9: Linha-Rec(a, t, e, x, n)

Entrada: $a, t \rightarrow \text{matrizes } 2 \times n$

 $e, x \rightarrow$ vetores de tamanho 2

 $n \rightarrow$ número de estações

Saída: Valor Ótimo

1 **criar vetor** $f_1[1 \cdots n]$, $f_2[1 \cdots n]$ com -1's

- 2 $f_1[1] \leftarrow e_1 + a_{1,1}$
- 3 $f_2[1] \leftarrow e_2 + a_{2,1}$
- 4 $z_1 \leftarrow \text{Linha-Rec-Rec}(1, n, a, t, e, x, n)$

Prof. Eurinardo

Aula Passada

Linha de Montagem

Evemple

Algoritme

Recursivo

Memoização
Programação Dinâmica

Linha de Montagem

Evennele

Algoritm

Pacureiua

Memoização

Programação Dinâmico

Algoritmo 10: Linha-Rec(a, t, e, x, n)

Entrada: $a, t \rightarrow \text{matrizes } 2 \times n$

 $e, x \rightarrow$ vetores de tamanho 2

 $n \rightarrow$ número de estações

Saída: Valor Ótimo

1 criar vetor $f_1[1 \cdots n]$, $f_2[1 \cdots n]$ com -1's

- 2 $f_1[1] \leftarrow e_1 + a_{1,1}$
- 3 $f_2[1] \leftarrow e_2 + a_{2,1}$
- 4 $z_1 \leftarrow \text{Linha-Rec-Rec}(1, n, a, t, e, x, n)$
- 5 $z_2 \leftarrow \text{Linha-Rec-Rec}(2, n, a, t, e, x, n)$

Linha de Montagem

Problem

Exemplo

Pacureius

Memoização

Programação Dinâmica

Algoritmo 11: Linha-Rec(a, t, e, x, n)

Entrada: $a, t \rightarrow \text{matrizes } 2 \times n$

 $e, x \rightarrow$ vetores de tamanho 2

 $n \rightarrow$ número de estações

Saída: Valor Ótimo

1 criar vetor $f_1[1 \cdots n]$, $f_2[1 \cdots n]$ com -1's

- 2 $f_1[1] \leftarrow e_1 + a_{1,1}$
- 3 $f_2[1] \leftarrow e_2 + a_{2,1}$
- 4 $z_1 \leftarrow \text{Linha-Rec-Rec}(1, n, a, t, e, x, n)$
- 5 $z_2 \leftarrow \text{Linha-Rec-Rec}(2, n, a, t, e, x, n)$
- 6 **retorne** min $\{z_1 + x_1, z_2 + x_2\}$

PAA - Aula 16

Prof. Eurinardo

Aula Passada

Linha de

Montagem Problema

Exemplo

Recursivo

Memoização

Algoritmo 13: Linha-Rec-Rec(I, k, a, t, e, x, n)

Entrada: $a, t \rightarrow \text{matrizes } 2 \times n$

PAA - Aula 16

Prof. Eurinardo

Aula Passada

Linha de Montagem

Problema

Exemple

Paguraina

Recursivo Memoização

Algoritmo 14: Linha-Rec-Rec(I, k, a, t, e, x, n)

Entrada: $a, t \rightarrow \text{matrizes } 2 \times n$ $e, x \rightarrow \text{vetor de tamanho } 2$ PAA - Aula 16

Prof. Eurinardo

Aula Passada

Linha de Montagem

Problema

Exempl

Recursivo

Memoização

Algoritmo 15: Linha-Rec-Rec(I, k, a, t, e, x, n)

Entrada: $a, t \rightarrow$ matrizes $2 \times n$ $e, x \rightarrow$ vetor de tamanho 2 $n \rightarrow$ número de estações PAA - Aula 16

Prof. Eurinardo

Aula Passada

Linha de Montagem

Exemple

Exempl

Recursivo

Recursivo Memoização

Algoritmo 16: Linha-Rec-Rec(I, k, a, t, e, x, n)

Entrada: $a, t \rightarrow \text{matrizes } 2 \times n$

 $e, x \rightarrow$ vetor de tamanho 2

 $n \rightarrow$ número de estações

 $I \rightarrow linha$

PAA - Aula 16

Prof. Eurinardo

Aula Passada

Linha de Montagem

Exemplo

Algoritme

Recursivo

Memoização
Programação Dinâmica

Algoritmo 17: Linha-Rec-Rec(I, k, a, t, e, x, n)

Entrada: $a, t \rightarrow \text{matrizes } 2 \times n$

 $e, x \rightarrow$ vetor de tamanho 2

 $n \rightarrow$ número de estações

 $I \rightarrow linha$

 $k \rightarrow \text{estação}$

PAA - Aula 16

Prof. Eurinardo

Aula Passada

Linha de Montagem

Evemblo

Almonitore

Recursivo

Memoização

Algoritmo 18: Linha-Rec-Rec(I, k, a, t, e, x, n)

Entrada: $a, t \rightarrow \text{matrizes } 2 \times n$

 $e, x \rightarrow$ vetor de tamanho 2

n o número de estações

 $I \rightarrow linha$

 $k \rightarrow \mathsf{esta}$ ção

Saída: Valor Ótimo para sair da (linha *l*, estação *k*)

PAA - Aula 16

Prof. Eurinardo

Aula Passada

Linha de Montagem

Exemplo

Algoritmo

Recursivo Memoização

Algoritmo 19: Linha-Rec-Rec(I, k, a, t, e, x, n)

Entrada: $a, t \rightarrow \text{matrizes } 2 \times n$

 $e, x \rightarrow$ vetor de tamanho 2

 $n \rightarrow$ número de estações

 $I \rightarrow linha$

k oestação

Saída: Valor Ótimo para sair da (linha *l*, estação *k*)

1 se / = 1 então

PAA - Aula 16

Prof. Eurinardo

Aula Passada

Linha de Montagem

Evenne

Exemplo

Recursivo

Memoização

```
Algoritmo 20: Linha-Rec-Rec(I, k, a, t, e, x, n)
```

```
Entrada: a, t \rightarrow matrizes 2 \times n

e, x \rightarrow vetor de tamanho 2

n \rightarrow número de estações

l \rightarrow linha

k \rightarrow estação
```

Saída: Valor Ótimo para sair da (linha *l*, estação *k*)

```
se / = 1 então
```

2 | se $f_1[k] \neq -1$ então

PAA - Aula 16

Prof. Eurinardo

Aula Passada

Linha de Montagem

Exemplo

Algoritmos

Recursivo Memoização

retorne $f_1[k]$

3

```
Exemplo
```

Algoritm

Recursivo

Memoização

Programação Dinâmica

```
Algoritmo 21: Linha-Rec-Rec(I, k, a, t, e, x, n)
Entrada: a, t \rightarrow matrizes 2 \times n
e, x \rightarrow vetor de tamanho 2
n \rightarrow número de estações
l \rightarrow linha
k \rightarrow estação
Saída: Valor Ótimo para sair da (linha l, estação k)

1 se l = 1 então
2 | se f_1[k] \neq -1 então
```

2

```
Algoritmo 22: Linha-Rec-Rec(I, k, a, t, e, x, n)
```

Aula Passada

```
Linha de
Montagem
Problema
Exemplo
```

Algoritmos Recursivo

Memoização
Programação Dinâmica

2

5

```
Algoritmo 23: Linha-Rec-Rec(I, k, a, t, e, x, n)
   Entrada: a, t \rightarrow \text{matrizes } 2 \times n
                  e, x \rightarrow \text{vetor de tamanho 2}
                  n \rightarrow número de estações
                  I \rightarrow linha
                  k \rightarrow \text{estação}
   Saída: Valor Ótimo para sair da (linha I, estação k)
   se l=1 então
          se f_1[k] \neq -1 então
3
                 retorne f_1[k]
           f_1[k] \leftarrow \min \left\{ \begin{array}{l} \text{Linha-Rec-Rec}(1, k-1) \\ \text{Linha-Rec-Rec}(2, k-1) + t_{2,k-1} \end{array} \right\} + a_{1,k}
          retorne f_1[k]
```

Prof. Eurinardo Aula Passada

```
Linha de
Montagem
```

Memoização

```
Algoritmo 24: Linha-Rec-Rec(I, k, a, t, e, x, n)
     Entrada: a, t \rightarrow \text{matrizes } 2 \times n
                    e. x \rightarrow \text{vetor de tamanho 2}
                    n \rightarrow número de estações
                    I \rightarrow linha
                    k \rightarrow \text{estação}
     Saída: Valor Ótimo para sair da (linha /, estação k)
     se l=1 então
            se f_1[k] \neq -1 então
 2
            retorne f_1[k]
             f_1[k] \leftarrow \min \left\{ \begin{array}{l} \text{Linha-Rec-Rec}(1, k-1) \\ \text{Linha-Rec-Rec}(2, k-1) + t_{2,k-1} \end{array} \right\} + a_{1,k}
 4
            retorne f_1[k]
     senão
            se f_2[k] \neq -1 então
             retorne f_2[k]
 8
             f_2[k] \leftarrow \min \left\{ \begin{array}{l} \text{Linha-Rec-Rec}(2, k-1) \\ \text{Linha-Rec-Rec}(1, k-1) + t_{1,k-1} \end{array} \right\} + a_{2,k}
 9
            retorne f_2[k]
10
```

Montagem
Problema
Exemplo
Algoritmos

Linha de

Memoização Programação Dinâmica

Linha de Montagem - PD

PAA - Aula 16

Prof. Eurinardo

Aula Passada

Linha de

Problema

Exemplo

Recursivo

Memoização

1 criar vetor $f_1[1\cdots n], f_2[1\cdots n];$

PAA - Aula 16

Prof. Eurinardo

Aula Passada

Linha de Montagem

Exemplo

Algorita

Recursivo

Memoização

Algoritmo 27: Linha-de-Motagem(a, t, e, x, n)

- 1 criar vetor $f_1[1 \cdots n], f_2[1 \cdots n];$
- 2 $f_1[1] \leftarrow e_1 + a_{1,1}$

PAA - Aula 16

Prof. Eurinardo

Aula Passada

Linha de Montagem

Algoritmo 28: Linha-de-Motagem(a, t, e, x, n)

- 1 criar vetor $f_1[1 \cdots n], f_2[1 \cdots n]$;
- 2 $f_1[1] \leftarrow e_1 + a_{1,1}$
- 3 $f_2[1] \leftarrow e_2 + a_{21}$

Prof. Eurinardo

Aula Passada

Linha de Montagem

Algoritmo 29: Linha-de-Motagem(a, t, e, x, n)

- 1 criar vetor $f_1[1 \cdots n], f_2[1 \cdots n];$
- 2 $f_1[1] \leftarrow e_1 + a_{1,1}$
- 3 $f_2[1] \leftarrow e_2 + a_{2,1}$
- 4 para $k \leftarrow 2$ até n faça

Prof. Eurinardo

Aula Passada

Linha de Montagem

Exemple

A. .

Recursivo

Algoritmo 30: Linha-de-Motagem(a, t, e, x, n)

Prof. Eurinardo

Aula Passada Linha de

Montagem Problema

Exemple

Aigoritmos

Recursivo

Algoritmo 31: Linha-de-Motagem(a, t, e, x, n)

```
1 criar vetor f_1[1\cdots n], f_2[1\cdots n];

2 f_1[1] \leftarrow e_1 + a_{1,1}

3 f_2[1] \leftarrow e_2 + a_{2,1}

4 para k \leftarrow 2 até n faça

5 extbf{se} f_1[k-1] \leq f_2[k-1] + f_{2,k-1} então

6 extbf{f_1[k]} \leftarrow f_1[k-1] + a_{1,k}
```

FAA - Aula 10

Prof. Eurinardo

Aula Passada

Linha de Montagem

Exemple

Pocureiuo

Algoritmo 32: Linha-de-Motagem(a, t, e, x, n)

```
1 criar vetor f_1[1 \cdots n], f_2[1 \cdots n];

2 f_1[1] \leftarrow e_1 + a_{1,1}

3 f_2[1] \leftarrow e_2 + a_{2,1}

4 para k \leftarrow 2 até n faça

5 | se f_1[k-1] \leq f_2[k-1] + f_{2,k-1} então

6 | f_1[k] \leftarrow f_1[k-1] + a_{1,k}

7 senão
```

1701 Maia 10

Prof. Eurinardo

Aula Passada

Linha de Montagem

Exempl

Recureiue

Algoritmo 33: Linha-de-Motagem(a, t, e, x, n)

```
1 criar vetor f_1[1\cdots n], f_2[1\cdots n];

2 f_1[1] \leftarrow e_1 + a_{1,1}

3 f_2[1] \leftarrow e_2 + a_{2,1}

4 para k \leftarrow 2 até n faça

5 se f_1[k-1] \leq f_2[k-1] + f_{2,k-1} então

6 f_1[k] \leftarrow f_1[k-1] + a_{1,k}

7 senão

8 f_1[k] \leftarrow f_2[k-1] + f_{2,k-1} + a_{1,k}
```

Prof. Eurinardo

Aula Passada

Linha de Montagem

Exempl

Recursivo

Algoritmo 34: Linha-de-Motagem(a, t, e, x, n)

```
1 criar vetor f_1[1 \cdots n], f_2[1 \cdots n];
 2 f_1[1] \leftarrow e_1 + a_{1,1}
 3 f_2[1] \leftarrow e_2 + a_{21}
 4 para k \leftarrow 2 até n faça
 5
          se f_1[k-1] < f_2[k-1] + t_{2k-1} então
                f_1[k] \leftarrow f_1[k-1] + a_{1k}
 6
 7
          senão
                f_1[k] \leftarrow f_2[k-1] + t_{2,k-1} + a_{1,k}
 8
          se f_2[k-1] \le f_1[k-1] + t_{1,k-1} então
 9
                f_2[k] \leftarrow f_2[k-1] + a_{2k}
10
11
          senão
                f_2[k] \leftarrow f_1[k-1] + t_{1,k-1} + a_{2,k}
12
```

Prof. Eurinardo

Aula Passada

Linha de Montagem

Exempl

Recureius

Algoritmo 35: Linha-de-Motagem(a, t, e, x, n)

```
1 criar vetor f_1[1 \cdots n], f_2[1 \cdots n];
 2 f_1[1] \leftarrow e_1 + a_{1,1}
 3 f_2[1] \leftarrow e_2 + a_{21}
 4 para k \leftarrow 2 até n faça
 5
          se f_1[k-1] < f_2[k-1] + t_{2k-1} então
                f_1[k] \leftarrow f_1[k-1] + a_{1k}
 6
 7
          senão
                f_1[k] \leftarrow f_2[k-1] + t_{2,k-1} + a_{1,k}
 8
          se f_2[k-1] \le f_1[k-1] + t_{1,k-1} então
 9
                f_2[k] \leftarrow f_2[k-1] + a_{2k}
10
11
          senão
                f_2[k] \leftarrow f_1[k-1] + t_{1,k-1} + a_{2,k}
12
13 se f_1[n] + x_1 < f_2[n] + x_2 então
```

Prof. Eurinardo

Aula Passada

Linha de Montagem

Algoritmo 36: Linha-de-Motagem(a, t, e, x, n)

```
1 criar vetor f_1[1 \cdots n], f_2[1 \cdots n];
 2 f_1[1] \leftarrow e_1 + a_{1,1}
 3 f_2[1] \leftarrow e_2 + a_{21}
 4 para k \leftarrow 2 até n faça
          se f_1[k-1] \le f_2[k-1] + t_{2,k-1} então
 5
                f_1[k] \leftarrow f_1[k-1] + a_{1k}
 6
 7
          senão
                f_1[k] \leftarrow f_2[k-1] + t_{2k-1} + a_{1k}
 8
          se f_2[k-1] \le f_1[k-1] + t_{1,k-1} então
 9
                f_2[k] \leftarrow f_2[k-1] + a_{2k}
10
11
          senão
                f_2[k] \leftarrow f_1[k-1] + t_{1,k-1} + a_{2,k}
12
13 se f_1[n] + x_1 < f_2[n] + x_2 então
14 | f^* \leftarrow f_1[n] + x_1
```

Prof. Eurinardo

Aula Passada

Linha de Montagem

Exemplo

Recursivo

Algoritmo 37: Linha-de-Motagem(a, t, e, x, n)

```
1 criar vetor f_1[1 \cdots n], f_2[1 \cdots n];
 2 f_1[1] \leftarrow e_1 + a_{1,1}
 3 f_2[1] \leftarrow e_2 + a_{21}
 4 para k \leftarrow 2 até n faça
          se f_1[k-1] \le f_2[k-1] + t_{2,k-1} então
 5
               f_1[k] \leftarrow f_1[k-1] + a_{1k}
 6
 7
          senão
               f_1[k] \leftarrow f_2[k-1] + t_{2k-1} + a_{1k}
 8
          se f_2[k-1] \le f_1[k-1] + t_{1,k-1} então
 9
                f_2[k] \leftarrow f_2[k-1] + a_2 k
10
11
          senão
                f_2[k] \leftarrow f_1[k-1] + t_{1,k-1} + a_{2,k}
12
13 se f_1[n] + x_1 < f_2[n] + x_2 então
    f^* \leftarrow f_1[n] + x_1
14
15 senão
```

Prof. Eurinardo

Aula Passada

Linha de Montagem

Exempl

Recursivo

Algoritmo 38: Linha-de-Motagem(a, t, e, x, n)

```
1 criar vetor f_1[1 \cdots n], f_2[1 \cdots n];
 2 f_1[1] \leftarrow e_1 + a_{1,1}
 3 f_2[1] \leftarrow e_2 + a_{21}
 4 para k \leftarrow 2 até n faça
          se f_1[k-1] \le f_2[k-1] + t_{2,k-1} então
 5
                f_1[k] \leftarrow f_1[k-1] + a_{1k}
 6
 7
          senão
                f_1[k] \leftarrow f_2[k-1] + t_{2k-1} + a_{1k}
 8
          se f_2[k-1] \le f_1[k-1] + t_{1,k-1} então
 9
                f_2[k] \leftarrow f_2[k-1] + a_2 k
10
11
          senão
                f_2[k] \leftarrow f_1[k-1] + t_{1,k-1} + a_{2,k}
12
13 se f_1[n] + x_1 < f_2[n] + x_2 então
    f^* \leftarrow f_1[n] + x_1
15 senão
    f^* \leftarrow f_2[n] + x_2
16
```

Prof. Eurinardo

Aula Passada

Linha de Montagem

Algoritmo 39: Linha-de-Motagem(a, t, e, x, n)

```
1 criar vetor f_1[1 \cdots n], f_2[1 \cdots n];
 2 f_1[1] \leftarrow e_1 + a_{1,1}
 3 f_2[1] \leftarrow e_2 + a_2
 4 para k \leftarrow 2 até n faça
          se f_1[k-1] \le f_2[k-1] + t_{2,k-1} então
 5
               f_1[k] \leftarrow f_1[k-1] + a_{1k}
 6
 7
          senão
               f_1[k] \leftarrow f_2[k-1] + t_{2k-1} + a_{1k}
 8
          se f_2[k-1] \le f_1[k-1] + t_{1,k-1} então
 9
                f_2[k] \leftarrow f_2[k-1] + a_2 k
10
11
          senão
                f_2[k] \leftarrow f_1[k-1] + t_{1,k-1} + a_{2,k}
12
13 se f_1[n] + x_1 < f_2[n] + x_2 então
    f^* \leftarrow f_1[n] + x_1
15 senão
    f^* \leftarrow f_2[n] + x_2
17 retorne f*
```

Prof. Eurinardo

Aula Passada

Linha de Montagem

Exemple

Recursivo

Algoritmo 40: Linha-de-Motagem(a, t, e, x, n)

```
1 criar vetor f_1[1 \cdots n], f_2[1 \cdots n], \ell_1[1 \cdots n], \ell_2[1 \cdots n];
 2 f_1[1] \leftarrow e_1 + a_{1,1}; \ell_1[1] \leftarrow 1;
 3 f_2[1] \leftarrow e_2 + a_{2,1}; \ell_2[1] \leftarrow 2;
 4 para k \leftarrow 2 até n faça
           se f_1[k-1] < f_2[k-1] + t_{2k-1} então
 5
                  f_1[k] \leftarrow f_1[k-1] + a_{1k}; \ell_1[k] \leftarrow 1;
 6
 7
           senão
                  f_1[k] \leftarrow f_2[k-1] + f_2[k-1] + a_1[k] + a_1[k] \leftarrow 2;
 8
           se f_2[k-1] < f_1[k-1] + t_{1,k-1} então
 9
                  f_2[k] \leftarrow f_2[k-1] + a_{2k}; \ell_2[k] \leftarrow 2;
10
           senão
11
                  f_2[k] \leftarrow f_1[k-1] + t_{1,k-1} + a_{2,k}; \ell_2[k] \leftarrow 1;
12
13 se f_1[n] + x_1 < f_2[n] + x_2 então
      f^* \leftarrow f_1[n] + x_1; \ell^* \leftarrow 1;
15 senão
     f^* \leftarrow f_2[n] + x_2; \ell^* \leftarrow 2;
17 retorne f^*, \ell_1, \ell_2, \ell^*;
```

Prof. Eurinardo

Aula Passada

Linha de Montagem

Linha de Montagem - PD

PAA - Aula 16

Prof. Eurinardo

Aula Passada

Linha de

Problema

Exemplo

Recursivo

Memoização

Linha de Montagem - PD

PAA - Aula 16

Prof. Eurinardo

Aula Passada

Linha de Montagem

Programação Dinâmica

Algoritmo 42: Imprimir(ℓ_1, ℓ_2, ℓ^*)

Problema

Exemplo

Recursivo

Memoização

Programação Dinâmica

Algoritmo 43: Imprimir (ℓ_1, ℓ_2, ℓ^*)

1 $i \leftarrow \ell^*$

Problema

Exemplo

Aigoritinos

Recursivo

Programação Dinâmica

Algoritmo 44: Imprimir(ℓ_1 , ℓ_2 , ℓ^*)

- 1 $i \leftarrow \ell^*$
- 2 escreva "linha" i "Estação" n

Aula Passada

Linha de Montagem

Problema

Exemple

Recursivo

Memoização

```
Algoritmo 45: Imprimir(\ell_1, \ell_2, \ell^*)
```

- 1 $i \leftarrow \ell^*$
- 2 escreva "linha" i "Estação" n
- 3 para j ← n até 2 faça

Aula Passada

Linha de Montagem

```
Problema
Exemplo
```

Exemplo

Recursivo

```
Algoritmo 46: Imprimir(\ell_1, \ell_2, \ell^*)
```

```
1 i \leftarrow \ell^*
```

- 2 escreva "linha" i "Estação" n
- 3 para j ← n até 2 faça
- 4 $i \leftarrow \ell_i[j]$

Algoritmo 47: Imprimir(ℓ_1, ℓ_2, ℓ^*)

```
1 i \leftarrow \ell^*
2 escreva "linha" i "Estação" n
3 para j ← n até 2 faça
    i \leftarrow \ell_i[j]
5 escreva "linha" i "Estação" j − 1
```

Prof. Eurinardo

Aula Passada

Linha de Montagem

Aula Passada

Linha de Montagem

Exempl

Algoritmos

Memoização

Memoização
Programação Dinâmica

LEISERSON, C.E., STEIN, C., RIVEST, R.L., CORMEN T.H.

Algoritmos: teoria e prática, 3ed. Editora Campus, ano 2012.

Aula Passada

Linha de Montagem

Exemple

Algoritm

Recursivo

Programação Dinâmica

Obrigado!