

Aula Prática 4

ASA 2022/2023

Q1 (T2 20/21): Uma sequência diz-se um *palíndromo* se é simétrica, isto é, se permanece igual quando lida de trás para diante; por exemplo, são palíndromos as sequências: a , aa , $abbba$ e $abbaabba$. Pretende-se desenvolver um algoritmo que, dada uma sequência de caracteres arbitrária, retorne o tamanho do maior palíndromo que esta contém. Por exemplo, dada a sequência $abbaabbabaabc$, o algoritmo deve retornar 8, que corresponde ao tamanho do palíndromo $abbaabba$.

1. Seja $x[1..n]$ a string de texto dada como input e $B(i, j)$ o valor Booleano que indica se a cadeia de caracteres $x[i..j]$ forma um palíndromo. Defina $B(i, j)$ recursivamente completando os campos em baixo:

$$B(i, j) = \begin{cases} \text{true} & \text{se } j < i \\ \boxed{\phantom{\text{true}}} & \text{se } j = i \\ \boxed{\phantom{\text{true}}} & \text{c.c.} \end{cases}$$

Admite-se, para simplificar a formulação, que $B(i, j) = \mathbf{true}$ quando $j < i$.

- Complete o template de código em baixo que calcula o tamanho do maior palíndromo contido no array dado como input, $x[1..n]$. Para obter a cotação máxima, o algoritmo deve retornar o valor pretendido assim que encontra o palíndromo de tamanho máximo, não devendo de efectuar o preenchimento completo da matriz $B[1..n, 1..n]$.

$$\text{BiggestPalindromeSize}(x[1..n])$$

let $B[1..n, 1..n]$ be a new matrix of size $n \times n$ with all cells initialised to true

Case No.	Case Name	Case Address	Case City	Case State	Case Zip	Case Phone	Case Email	Case Date	Case Time	Case Status	Case Notes
1	John Doe	123 Main St	New York	NY	10001	212-555-1234	john.doe@example.com	2023-10-27	14:30	Completed	Initial assessment and data collection.
2	Jane Smith	456 Elm St	Los Angeles	CA	90001	310-555-5678	jane.smith@example.com	2023-10-28	10:00	In Progress	Interview with subject and review of records.
3	Robert Johnson	789 Oak St	Chicago	IL	60601	312-555-9012	robert.johnson@example.com	2023-10-29	09:00	Pending	Awaiting approval for field visit.
4	Maria Garcia	101 Pine St	San Francisco	CA	94101	415-555-3456	maria.garcia@example.com	2023-10-30	11:00	Completed	Final report and analysis completed.
5	David Lee	202 Birch St	Seattle	WA	98101	206-555-7890	david.lee@example.com	2023-10-31	13:00	In Progress	Conducting follow-up interviews.
6	Emily White	303 Cedar St	Portland	OR	97201	503-555-2345	emily.white@example.com	2023-11-01	15:00	Pending	Waiting for data from other sources.
7	Michael Brown	404 Maple St	Denver	CO	80201	303-555-6789	michael.brown@example.com	2023-11-02	10:30	Completed	Interview and data collection complete.
8	Sarah Davis	505 Spruce St	Phoenix	AZ	85001	602-555-0123	sarah.davis@example.com	2023-11-03	12:00	In Progress	Reviewing collected data and reports.
9	James Wilson	606 Ash St	San Diego	CA	92101	619-555-4567	james.wilson@example.com	2023-11-04	14:00	Pending	Awaiting final review and approval.
10	Lisa Anderson	707 Hickory St	San Jose	CA	95101	408-555-8901	lisa.anderson@example.com	2023-11-05	11:30	Completed	Final report and analysis completed.

3. Determine a complexidade assintótica do algoritmo proposto na alínea anterior.

Q2 (R2 20/21): Dada uma sequência de inteiros positivos $\langle x_1, \dots, x_n \rangle$, pretende desenvolver-se um algoritmo que determina o maior valor susceptível de ser obtido a partir da expressão $x_1/x_2/x_3/\dots/x_n$, determinando a ordem pela qual as divisões devem ser efectuadas. Por exemplo, dada a sequência $\langle 16, 8, 4, 2 \rangle$, a parentização que resulta no maior valor final é: $(16/((8/4)/2)) = 16$.

1. Seja $M[i, j]$ o maior valor que é possível obter a partir da expressão $x_i/x_{i+1}/\dots/x_j$ e $m[i, j]$ o menor valor. Por exemplo, dada a sequência $\langle 16, 8, 4, 2 \rangle$, $M[1, 4] = 16$ e $m[1, 4] = 0.25$. Admitindo que a sequência dada como input é $\langle x_1, \dots, x_n \rangle$, defina $M[i, j]$ e $m[i, j]$ recursivamente completando os campos em baixo:

$$M(i, j) = \begin{cases} \boxed{} & \text{se } i = j \\ \boxed{} & \text{se } j > i \end{cases}$$

$$m(i, j) = \begin{cases} \boxed{} & \text{se } j = i \\ \boxed{} & \text{se } j > i \end{cases}$$

2. Complete o template de código em baixo que, dada uma sequência de inteiros $\langle x_1, \dots, x_n \rangle$, calcula $m[1, n]$ e $M[1, n]$.

GreatestValue($x[1..n]$)

let $M[1..n, 1..n]$ **be** a new matrix of size $n \times n$

let $m[1..n, 1..n]$ **be** a new matrix of size $n \times n$

for $i = 1$ **to** n **do**

$M[i, i] := \boxed{}$

$m[i, i] := \boxed{}$

endfor

for $s = 1$ **to** $n-1$ **do**

for $i = 1$ **to** $n-s$ **do**

endfor

endfor

return $M[1, n]$

3. Determine a complexidade assintótica do algoritmo proposto na alínea anterior.

Q3 (EE 20/21): Dadas duas seqüências de caracteres $\vec{X} = \langle X_1, \dots, X_n \rangle$ e $\vec{Z} = \langle Z_1, \dots, Z_k \rangle$, \vec{Z} diz-se uma *subseqüência contígua* de \vec{X} se existir um inteiro $0 \leq i < n$ tal que: $X_{i+1} = Z_1, X_{i+2} = Z_2, \dots, X_{i+k} = Z_k$. Por exemplo, a seqüência de caracteres *abb* é uma subseqüência contígua de *ababb* (basta escolher o deslocamento $i = 2$).

Dadas duas seqüências de caracteres $\vec{X} = \langle X_1, \dots, X_n \rangle$ e $\vec{Y} = \langle Y_1, \dots, Y_m \rangle$, pretende desenvolver-se um algoritmo que determine o tamanho da sua maior subseqüência contígua comum.

1. Dadas duas seqüências de caracteres $\vec{X} = \langle X_1, \dots, X_n \rangle$ e $\vec{Y} = \langle Y_1, \dots, Y_m \rangle$, seja $B(i, j)$ o tamanho do maior sufixo comum entre $\langle X_1, \dots, X_i \rangle$ e $\langle Y_1, \dots, Y_j \rangle$. Por exemplo, para $\vec{X} = abaabb$ e $\vec{Y} = abbbbb$, temos que $B(3, 3) = 0$ e $B(6, 3) = 3$. Defina $B(i, j)$ recursivamente completando os campos em baixo:

$$B(i, j) = \begin{cases} 0 & \text{se } i = 0 \vee j = 0 \\ \boxed{} & \text{se } \boxed{} \\ \boxed{} & \text{c.c.} \end{cases}$$

Admite-se, para simplificar a formulação, que $B(i, j) = 0$ quando $i = 0$ ou $j = 0$.

2. Complete o template de código em baixo que, dadas duas seqüências de caracteres $\langle X_1, \dots, X_n \rangle$ e $\langle Y_1, \dots, Y_m \rangle$, calcula o tamanho da sua maior subseqüência contígua comum.

```

LongestContiguousCommonSubstring( $x[1..n], y[1..m]$ )
  let  $B[0..n, 0..m]$  be a new matrix of size  $(n + 1) \times (m + 1)$ 
   $B[0, 0] := \boxed{\phantom{000000}}$ 
  for  $i = 1$  to  $n$  do
     $B[i, 0] := \boxed{\phantom{000000}}$ 
  endfor
  for  $j = 1$  to  $m$  do
     $B[0, j] := \boxed{\phantom{000000}}$ 
  endfor
  let  $max = 0$ 
  for  $i = 1$  to  $n$  do
    for  $j = 1$  to  $m$  do
      
    endfor
  endfor
  return  $max$ 

```

3. Determine a complexidade assintótica do algoritmo proposto na alínea anterior.

Q4 (T2 08/09 II.2) Considere o problema de determinar a colocação óptima de parêntesis, que permite reduzir o número de operações na multiplicação de matrizes. Como sabe, o número de operações mínimo para efectuar a multiplicação $A_i A_{i+1} \dots A_j$ é dado por:

$$m[i, j] = \begin{cases} 0 & \text{if } i = j \\ \min_{i \leq k < j} \{m[i, k] + m[k + 1, j] + p_{i-1}p_kp_j\} & \text{if } i < j \end{cases}$$

Considerando as matrizes A , B , C e D com as seguintes dimensões:

Matriz	Dimensão
A	2×5
B	5×3
C	3×1
D	1×2

Indique qual a colocação óptima de parêntesis para o produto $ABCD$. Para o efeito deverá escrever a expressão do produto $ABCD$, colocando os parêntesis na posição correcta. Adicionalmente, indique os valores de $m[1, 2]$, $m[1, 4]$, $m[1, 3]$ e $m[2, 4]$.

Q5 (R2 08/09 II.2) Considere o problema da identificação da maior subsequência comum (LCS) entre duas sequências, S e T . Admita que, numa formulação do problema em termos de programação dinâmica, o comprimento da maior subsequência comum entre os prefixos $S_i = \langle s_1, s_2, \dots, s_i \rangle$ e $T_j = \langle t_1, t_2, \dots, t_j \rangle$ é definido por:

$$c[i, j] = \begin{cases} 0 & \text{se } i = 0 \vee j = 0 \\ c[i - 1, j - 1] + 1 & \text{se } i, j > 0 \wedge s_i = t_j \\ \max(c[i - 1, j], c[i, j - 1]) & \text{se } i, j > 0 \wedge s_i \neq t_j \end{cases}$$

Dadas as sequências $S = ABCBCDBBDCABCDB$ e $T = ABBACBDCCDBACD$, indique qual a LCS, bem como os seguintes valores: $c[0, 10]$, $c[4, 6]$, $c[5, 12]$, $c[9, 13]$, $c[10, 10]$, $c[14, 14]$ e $c[15, 14]$.