Midterm practice test

Instructor: Gleb Pogudin
Consider
\[ f(n) = 317 + 5n^2 + n \log_2(n) + \log_2 n. \]

Which of the following is false?

(A) \( f(n) = O(n^2 \log_2 n) \)
(B) \( f(n) = O(n^2 + n \log_2 n) \)
(C) \( f(n) = O(n \log_2(n^2)) \)
(D) \( f(n) = O(n^2 + \log_2 n) \)

Answer: C
Problem 1

Consider

\[ f(n) = 317 + 5n^2 + n \log_2(n) + \log_2 n = \Theta(n^2). \]

Which of the following is false?

(A) \( f(n) = O(n^2 \log_2 n) \)
(B) \( f(n) = O(n^2 + n \log_2 n) \)
(C) \( f(n) = O(n \log_2(n^2)) \)
(D) \( f(n) = O(n^2 + \log_2 n) \)

Answer: C
Problem 1

Consider

\[ f(n) = 317 + 5n^2 + n \log_2(n) + \log_2 n = \Theta(n^2). \]

Which of the following is false?

(A) \( f(n) = O(n^2 \log_2 n) \)
(B) \( f(n) = O(n^2 + n \log_2 n) = O(n^2) \)
(C) \( f(n) = O(n \log_2(n^2)) \)
(D) \( f(n) = O(n^2 + \log_2 n) \)

Answer: C
Consider

\[ f(n) = 317 + 5n^2 + n \log_2(n) + \log_2 n = \Theta(n^2). \]

Which of the following is false?

(A) \( f(n) = O(n^2 \log_2 n) \)
(B) \( f(n) = O(n^2 + n \log_2 n) = O(n^2) \)
(C) \( f(n) = O(n \log_2(n^2)) = O(n \log_2 n) \) (because \( \log_2(n^2) = 2 \log_2 n \))
(D) \( f(n) = O(n^2 + \log_2 n) \)

Answer: C
Consider

\[ f(n) = 317 + 5n^2 + n \log_2(n) + \log_2 n = \Theta(n^2). \]

Which of the following is false?

(A) \( f(n) = O(n^2 \log_2 n) \)

(B) \( f(n) = O(n^2 + n \log_2 n) = O(n^2) \)

(C) \( f(n) = O(n \log_2(n^2)) = O(n \log_2 n) \) (because \( \log_2(n^2) = 2 \log_2 n \))

(D) \( f(n) = O(n^2 + \log_2 n) = O(n^2) \)
Consider

\[ f(n) = 317 + 5n^2 + n \log_2(n) + \log_2 n = \Theta(n^2). \]

Which of the following is false?

(A) \( f(n) = O(n^2 \log_2 n) \)
(B) \( f(n) = O(n^2 + n \log_2 n) = O(n^2) \)
(C) \( f(n) = O(n \log_2(n^2)) = O(n \log_2 n) \) (because \( \log_2(n^2) = 2 \log_2 n \))
(D) \( f(n) = O(n^2 + \log_2 n) = O(n^2) \)

**Answer:** C
Problem 2

Function \textbf{Mystery}(n)
\begin{align*}
answer &= 0 \\
\text{for } i \text{ from } 1 \text{ to } n \text{ do} \\
&\quad \text{for } j \text{ from } 1 \text{ to } i \text{ do} \\
&\quad \quad answer = answer + i \\
&\quad \text{end for} \\
&\text{end for} \\
\text{while } answer > 0 \text{ do} \\
&\quad answer = answer - 1 \\
\text{end while} \\
\text{EndFunction}
\end{align*}

Out of the following, which is the best possible bound we can give on the running time of \textbf{Mystery}(n)?

(A) $O(n^3 \log_2 n)$
(B) $O(n^2 \log_2 n)$
(C) $O(n^2)$
(D) $O(n^3)$

Answer: D
Function Mystery(n)
answer = 0
for i from 1 to n do
    for j from 1 to i do
        answer = answer + i
    end for
end for
// answer = 1^2 + 2^2 + ... + n^2
while answer > 0 do
    answer = answer – 1
end while
EndFunction

Out of the following, which is the best possible bound we can give on the running time of Mystery(n)?

(A) $O(n^3 \log_2 n)$
(B) $O(n^2 \log_2 n)$
(C) $O(n^2)$
(D) $O(n^3)$
Problem 2

Function **Mystery**(*n*)

\[ \text{answer} = 0 \]

\[ \text{for } i \text{ from } 1 \text{ to } n \text{ do} \]
\[ \quad \text{for } j \text{ from } 1 \text{ to } i \text{ do} \]
\[ \quad \quad \text{answer} = \text{answer} + i \]
\[ \quad \text{end for} \]
\[ \text{end for} \]

\[ \text{while } \text{answer} > 0 \text{ do} \]
\[ \quad \text{answer} = \text{answer} - 1 \]
\[ \text{end while} \]
EndFunction

Out of the following, which is the best possible bound we can give on the running time of **Mystery**(*n*)?

(A) \( O(n^3 \log_2 n) \)
(B) \( O(n^2 \log_2 n) \)
(C) \( O(n^2) \)
(D) \( O(n^3) \)

**Answer:** D
Problem 3

Function Smth(T)
if T = NULL then
    print “A”
    return 0
end if
r = 0
for C in T.children do
    if T.data < C.data then
        print “B”
        r = r + 1
    end if
    r = r + Smth(C)
end for
print “C”
return r
EndFunction

“A” will be printed ??? times
“B” will be printed ??? times
“C” will be printed ??? times
Function Smth(T)
if T = NULL then
    print “A”
    return 0
end if
r = 0
for C in T.children do
    if T.data < C.data then
        print “B”
        r = r + 1
    end if
    r = r + Smth(C)
end for
print “C”
return r
EndFunction

“A” will be printed 0 times
“B” will be printed ??? times
“C” will be printed ??? times
Function `Smth(T)`

```
if T = NULL then
    print “A”
    return 0
end if

r = 0
for C in T.children do
    if T.data < C.data then
        print “B”
        r = r + 1
    end if
    r = r + Smth(C)
end for
print “C”
return r
EndFunction
```
Function `Smth(T)`

if `T` = NULL then
    print “A”
    return 0
end if

r = 0

for `C` in `T`.children do
    if `T`.data < `C`.data then
        print “B”
        r = r + 1
    end if
    r = r + Smth(C)
end for

print “C”
return r

EndFunction

“A” will be printed 0 times
“B” will be printed ??? times
“C” will be printed 13 times
Problem 3

Function Smth(T)
if $T = \text{NULL}$ then
    print “A”
    return 0
end if
$r = 0$
for $C$ in $T\.children$ do
    if $T\.data < C\.data$ then
        print “B”
        $r = r + 1$
    end if
    $r = r + \text{Smth}(C)$
end for
print “C”
return $r$
EndFunction

“A” will be printed 0 times
“B” will be printed 5 times
“C” will be printed 13 times
Function `Perm(L)`

if $L$ is empty then
  return [ [ ] ]
end if

$R = [ ]$

for $x$ in $L$ do
  for $p$ in $Perm(L - x)$ do
    append $[x || p]$ to $R$
  end for
end for

return $R$

EndFunction

Consider $Perm([1, 2, 3, 4])$

- The 4-th element of the output: ???
- The 10-th element of the output: ???
Function \texttt{Perm}(L)
if \( L \) is empty then
    return \([ [] ]\)
end if
\( R = [] \)
for \( x \) in \( L \) do
    for \( p \) in \texttt{Perm}(L - x) do
        append \([x||p]\) to \( R \)
    end for
end for
return \( R \)
EndFunction

Consider \texttt{Perm}([1, 2, 3, 4])

- The 4-th element of the output: 1, 3, 4, 2
- The 10-th element of the output: ???
Consider $\text{Perm}([1, 2, 3, 4])$

- The 4-th element of the output: 1, 3, 4, 2
- The 10-th element of the output: 2, 3, 4, 1
Problem 5

The following is the final state of the lookup table for computing the Longest Common Subsequence of “cabac” and some other string s. What is the longest common subsequence?

<table>
<thead>
<tr>
<th>i/j</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>c</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>a</td>
<td>2</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>b</td>
<td>3</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td>-1</td>
<td>2</td>
</tr>
<tr>
<td>a</td>
<td>4</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>c</td>
<td>5</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>4</td>
</tr>
</tbody>
</table>

Answer: “abac”
Problem 5

The following is the final state of the lookup table for computing the Longest Common Subsequence of "cabac" and some other string s. What is the longest common subsequence?

<table>
<thead>
<tr>
<th>i/j</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>c</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>a</td>
<td>2</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>b</td>
<td>3</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td>-1</td>
<td>2</td>
</tr>
<tr>
<td>a</td>
<td>4</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>c</td>
<td>5</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>4</td>
</tr>
</tbody>
</table>

Answer: “abac”.
Problem 5

The following is the final state of the lookup table for computing the Longest Common Subsequence of “cabac” and some other string $s$. What is the longest common subsequence?

<table>
<thead>
<tr>
<th>i/j</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>a</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>b</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td>-1</td>
<td>2</td>
<td>-1</td>
</tr>
<tr>
<td>a</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>3</td>
<td>3</td>
<td>-1</td>
</tr>
<tr>
<td>c</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>4</td>
</tr>
</tbody>
</table>

Answer: “abac”.

The following is the final state of the lookup table for computing the Longest Common Subsequence of “cabac” and some other string $s$. What is the longest common subsequence?

<table>
<thead>
<tr>
<th>i/j</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>−1</td>
<td>0</td>
<td>0</td>
<td>−1</td>
<td>−1</td>
</tr>
<tr>
<td>c</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>−1</td>
<td>−1</td>
</tr>
<tr>
<td>a</td>
<td></td>
<td>−1</td>
<td>1</td>
<td>−1</td>
<td>1</td>
<td>−1</td>
</tr>
<tr>
<td>b</td>
<td></td>
<td>−1</td>
<td>−1</td>
<td>2</td>
<td>−1</td>
<td>2</td>
</tr>
<tr>
<td>a</td>
<td></td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>c</td>
<td></td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
</tr>
</tbody>
</table>
The following is the final state of the lookup table for computing the Longest Common Subsequence of “cabac” and some other string s.

What is the longest common subsequence?

<table>
<thead>
<tr>
<th>i/j</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>c</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>a</td>
<td>2</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>b</td>
<td>3</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td>-1</td>
<td>2</td>
</tr>
<tr>
<td>a</td>
<td>4</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>c</td>
<td>5</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>4</td>
</tr>
</tbody>
</table>

Answer: “abac”. 

The following is the final state of the lookup table for computing the Longest Common Subsequence of “cabac” and some other string s. What is the longest common subsequence?

<table>
<thead>
<tr>
<th>i/j</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>a</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>b</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td>-1</td>
<td>2</td>
<td>-1</td>
</tr>
<tr>
<td>a</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>3</td>
<td>3</td>
<td>-1</td>
</tr>
<tr>
<td>c</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>4</td>
</tr>
</tbody>
</table>
Problem 5

The following is the final state of the lookup table for computing the Longest Common Subsequence of “cabac” and some other string s. What is the longest common subsequence?

<table>
<thead>
<tr>
<th>i/j</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>c</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>a</td>
<td>2</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>b</td>
<td>3</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td>-1</td>
<td>2</td>
</tr>
<tr>
<td>a</td>
<td>4</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>c</td>
<td>5</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>4</td>
</tr>
</tbody>
</table>

Answer: “abac”.
The following is the final state of the lookup table for computing the Longest Common Subsequence of “cabac” and some other string $s$. What is the longest common subsequence?

<table>
<thead>
<tr>
<th>i/j</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>a</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>b</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td>-1</td>
<td>2</td>
<td>-1</td>
</tr>
<tr>
<td>a</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>3</td>
<td>3</td>
<td>-1</td>
</tr>
<tr>
<td>c</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>4</td>
</tr>
</tbody>
</table>

**Answer:** “abac”.
Complete the final state of the lookup table for $\text{CoinsDP}(10, [1, 4, 7])$.

<table>
<thead>
<tr>
<th>$i$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>results[$i$]</td>
<td>0</td>
<td>1</td>
<td>?</td>
<td>3</td>
<td>?</td>
<td>2</td>
<td>?</td>
<td>1</td>
<td>2</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
Complete the final state of the lookup table for CoinsDP(10, [1, 4, 7]).

<table>
<thead>
<tr>
<th>$i$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>results[$i$]</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>?</td>
<td>2</td>
<td>?</td>
<td>1</td>
<td>2</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
Complete the final state of the lookup table for CoinsDP(10, [1, 4, 7]).

<table>
<thead>
<tr>
<th>$i$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>results[$i$]</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>?</td>
<td>1</td>
<td>2</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
Complete the final state of the lookup table for \( \text{CoinsDP}(10, [1, 4, 7]) \).

<table>
<thead>
<tr>
<th>( i )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>results[( i )]</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
Problem 6

Complete the final state of the lookup table for CoinsDP(10, [1, 4, 7]).

<table>
<thead>
<tr>
<th>$i$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>results[$i$]</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>?</td>
</tr>
</tbody>
</table>
Complete the final state of the lookup table for CoinsDP(10, [1, 4, 7]).

<table>
<thead>
<tr>
<th>i</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>results[i]</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>
Problem 6

Complete the final state of the lookup table for CoinsDP(10, [1, 4, 7]).

<table>
<thead>
<tr>
<th>$i$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>results[$i$]</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>
Problem 7

Fill the blanks in the final state of the lookup table for computing the edit distance between “dear” and “dare” using recursive DP.

<table>
<thead>
<tr>
<th>i/j</th>
<th>d</th>
<th>a</th>
<th>r</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Problem 7

Fill the blanks in the final state of the lookup table for computing the edit distance between “dear” and “dare” using recursive DP.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>d</th>
<th>a</th>
<th>r</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>i/j</td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>d</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>e</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>a</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>r</td>
<td>4</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>a</td>
<td>5</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
</tr>
</tbody>
</table>