Intermediate Data Structures & Algorithms – CS 141 (Discussion)

Amirali Darvishzadeh
Fall 2015
A subsequence is *palindromic* if it is the same whether read left to right or right to left. For the sequence


Has many palindromic subsequence, including \( A, C, G, C, A \) and \( A, A, A, A \).

Devise an algorithm that takes a sequence \( x[1 \ldots n] \) and returns the longest palindromic subsequence. Its running time should be \( O(n^2) \)
**Question 2**

**Edit distance:** A natural measure of the distance between two strings is the extent to which they can be *aligned*, or matched up. Technically, an alignment is simply a way of writing the strings one above the other. For instance, here are two possible alignments of SNOWY and SUNNY:

```
SNOWY
SUNNY
```

Cost: 3

```
SNOWY
SUNNY
```

Cost: 5

Given two strings str1 and str2 and operations (Insert, Remove, Replace) that can performed on str1. Find minimum number of edits (operations) required to convert ‘str1’ into ‘str2’.

Edit distance is so named because it can also be thought of as the minimum number of edits - insertions, deletions, and substitutions of characters - needed to transform the first string into the second.
Sequence similarity measure. Let $x$ and $y$ be two given DNA sequences, represented as strings with characters in the set \{A, G, C, T\}. The similarity measure of $x$ and $y$ is defined as the maximum score of any alignment of $x$ and $y$, where the score for an alignment is computed by adding substitution scores and deletion and insertion scores, as explained below. (Some operations have negative scores.)

The score for changing a character $x_i$ into a character $y_j$ is $\delta(x_i, y_j)$, where $\delta(\cdot, \cdot)$ is defined as follows:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>G</th>
<th>C</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.0</td>
<td>-0.1</td>
<td>-0.1</td>
<td>-0.15</td>
</tr>
<tr>
<td>G</td>
<td>1.0</td>
<td>-0.15</td>
<td>-0.1</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1.0</td>
<td></td>
<td>-0.1</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td></td>
<td></td>
<td>1.0</td>
<td></td>
</tr>
</tbody>
</table>

The score for deleting any letter $x_i$ (which we think of as changing $x_i$ into a ‘-’) is $\delta(x_i, -) = -0.2$. The score for inserting any $y_j$ is the same. For example, for the strings “AACCTGACATCTT” and “CCAGCGTCAACTT”, the alignment

\[
\begin{array}{cccccccccccc}
|   |   |   |   |   |   |   |   | | | | | | | | |
|   |   |   |   |   |   |   |   | | | | | | | | |
\end{array}
\]

will have similarity score

\[-0.2 - 0.2 + 1.0 + 1.0 - 0.15 + 1.0 - 0.2 + 1.0 - 0.2 + 1.0 + 0.2 + 1.0 + 1.0 + 1.0.

Your should use a modification of the edit-distance algorithm to compute the alignment with maximum similarity score.
b) Develop a program (in Python) that performs DNA sequence alignment, using dynamic programming.

• Input: two DNA sequences;
• Output:
  • A) the similarity score
  • B) Sequence alignment