Name, SURNAME and ID ⇒ SOLUTIONS

Middle East Technical University
Department of Computer Engineering

CENG 465

Introduction to Bioinformatics

Spring '2005-2006

Midterm Exam I

- **Duration: 100** minutes.
- **Exam:**
  - This is a **closed book**, **closed notes** exam. The use of any reference material is strictly forbidden.
  - No attempts of cheating will be tolerated. In case such attempts are observed, the students who took part in the act will be prosecuted. The legal code states that students who are found guilty of cheating shall be expelled from the university for **a minimum of one semester**!
- **About the exam questions:**
  - The points assigned for each question are shown in parenthesis next to the question.
  - For *True-False* type questions, put your results in the boxes provided.
- **This exam consists of *8* pages including this page. Check that you have them all!**
- **GOOD LUCK !**

Question 1	20
Question 2	20
Question 3	25
Question 4	25
Question 5	10
Total ⇒	100

$\boxed{1}$ **(20 pts)**

For the following 10 statements, indicate whether the statement is *true* or *false* by marking the corresponding box with **T** or **F**, respectively (2 points each).

i. In a sequence alignment, the higher the *p-value* is, the more significant is the alignment, i.e., the more biologically related are the sequences. F

ii. It is more likely to find a match for a DNA pattern of length 10 in the mouse genome by chance, than to find a match for a peptide (i.e., short protein) pattern of the same length in the mouse proteome. T

iii. If we use same gap opening and gap extension penalties in the *affine gap model*, e.g., -2 for gap opening and -2 for gap extension, the effect will be the same as using a *linear gap model.* T

iv. With the help of BLAST sequence search, one can get more *accurate* alignments compared to running Smith-Waterman between the query sequence and every database sequence. F

v. In local sequence alignment, the maximum score is always at the lower-right corner of the partial scores table. F

vi. Each amino acid in a protein is usually encoded by a tri-nucleotide (i.e., a nucleotide sequence of length three); however, sometimes an amino acid may be encoded by four nucleotides for robustness. F

vii. The root of a suffix tree may have more than two child nodes. T

viii. The number of leaf nodes in a suffix tree is equal to the number of suffixes of the string for which the tree is built. T

ix. The existence check of a pattern of length 4 in a suffix tree with 150 leaves is 10 times faster than in a suffix tree with 1500 leaves. F

x. The *sum of pairs score* of a multiple alignment can be lower than the total score of the pairwise Smith-Waterman alignments between all sequence pairs. T

2 (20 pts)

(a)(10 pts) Fill out the dynamic programming table for determining the optimum **global alignment** between the sequences GCTA and CCA. Assume that a match is scored +3 and that mismatches and spaces are penalized -1 each.

	-	G	C	T	A
-	0	-1	-2	-3	-4
C	-1	-1	2	1	0
C	-2	-2	2	1	0
A	-3	-3	1	1	4

(b)(10 pts) What is the optimum alignment corresponding to the table in part (a) and what is its score?

Optimum alignment not unique: (Two alignments with same max. score)

alignment 1:

```
G C T A      Score = 4
  |   |
- C C A
```

alignment 2:

```
G C T A      Score = 4
  |   |
C C - A
```

3 **(25 pts)**

Consider the following 4 DNA sequences and the optimum global alignment between all pairs, where match score is 3 and mismatch score is -2. A linear gap model with a gap penalty of -2 is used.

Sequences:

- GCAT
- GCCA
- GGCAT
- GGCA

Optimum global pairwise alignments and their corresponding scores:

1.
   ```
   GGCAT
   | |||
   G-CAT
   ```
   Score: 10

2.
   ```
   GGCA-
   | ||
   G-CAT
   ```
   Score: 5

3.
   ```
   GGCA
   | ||
   GCCA
   ```
   Score: 7

4.
   ```
   G-CAT
   | ||
   GCCA-
   ```
   Score: 5

5.
   ```
   GGCAT
   | ||
   GCCA-
   ```
   Score: 5

6.
   ```
   GGCAT
   ||||
   GGCA-
   ```
   Score: 10

(a)(10 pts) Suppose that we want a multiple alignment of these four sequences using the **Star Alignment** technique. Which one of these sequences would be the *center* sequence? Why? Show your numerical calculations.

GGCAT is the center sequence; because, it is the most similar sequence to all other sequences.

Total similarity score of GGCAT alignments $= 25 = (10+5+10) \boxed{\rightarrow \text{max}}$

" " " " GCAT " $= 20 = (10+5+5)$

" " " " GCCA " $= 17 = (7+5+5)$

" " " " GGCA " $= 22 = (5+7+10)$

(b)(10 pts) Using the progressive star alignment technique and the center sequence you have chosen in part (a), construct a multiple alignment of these four sequences. Show individual steps of construction (~~4~~ 3 steps actually if you don't count selection of center steps in total).

The star alignment technique uses the pairwise alignments to the center sequence to construct the multiple alignment step by step. (The order of steps may vary)

Step 1: Add GCAT

```
GGCAT
G-CAT
```

Step 2: Add GCCA

```
GGCAT
G-CAT
GCCA-
```

Step 0: Start the multiple alignment with the center sequence

```
GGCAT
```

Step 3: Add GGCA

```
GGCAT
G-CAT
GCCA-
GGCA-
```

} final multiple alignment

(c)(5 pts) What is the *sum of pairs score* of the alignment you constructed in part (b)?

Sum of pairs score is the sum of all pairwise alignments induced by the multiple alignment

$$\text{Sum of pairs} = S(GGCAT, GCAT) + S(GGCAT, GCCA) + \\ + S(GGCAT, GGCA) + S(GCAT, GCCA) + \\ + S(GCAT, GGCA) + S(GCCA, GGCA)$$

where $S(x,y)$ is computed by looking at the positions of letters in the multiple alignment

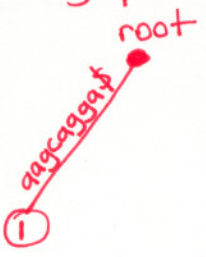
$$\Rightarrow \text{Sum of pairs} = 10 + 5 + 10 + 5 + 5 + 7 = \underline{\underline{42}}$$

4 (25 pts)

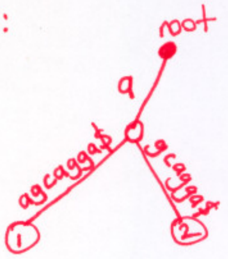
(a)(15 pts) Construct a suffix tree for the following string: **aagcagga$**. Show the individual steps of construction (9 steps in total).

Starting from the first suffix, we add suffixes one by one.

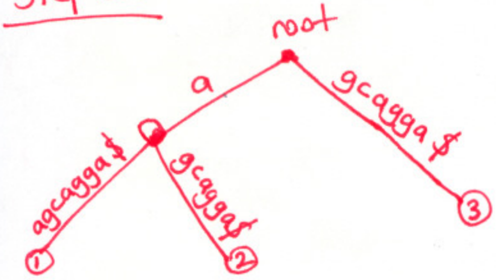
Step 1:



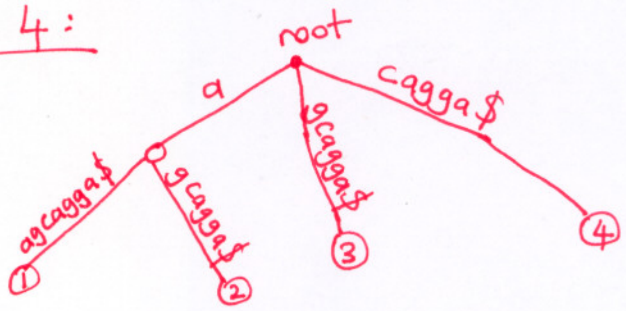
step 2:



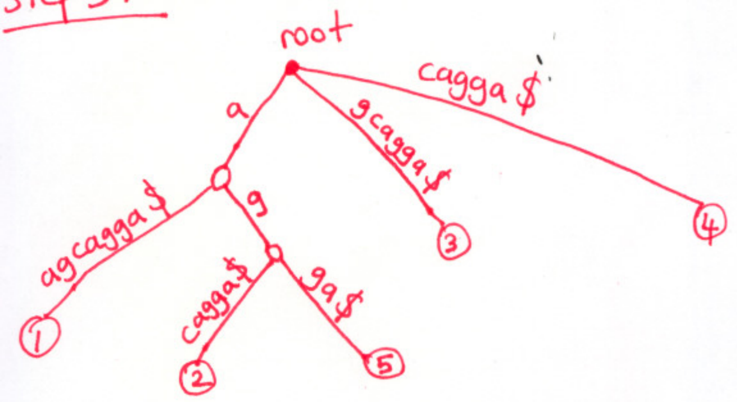
step 3:



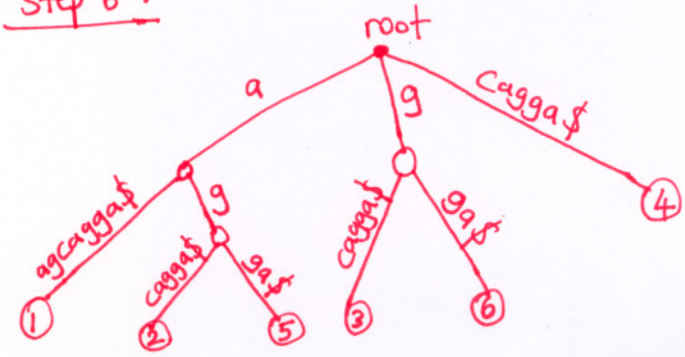
step 4:



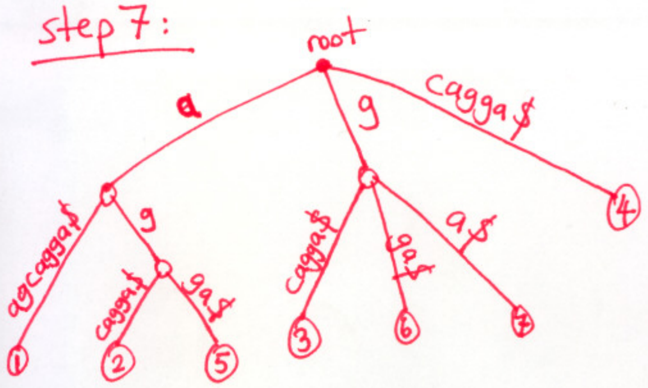
step 5:



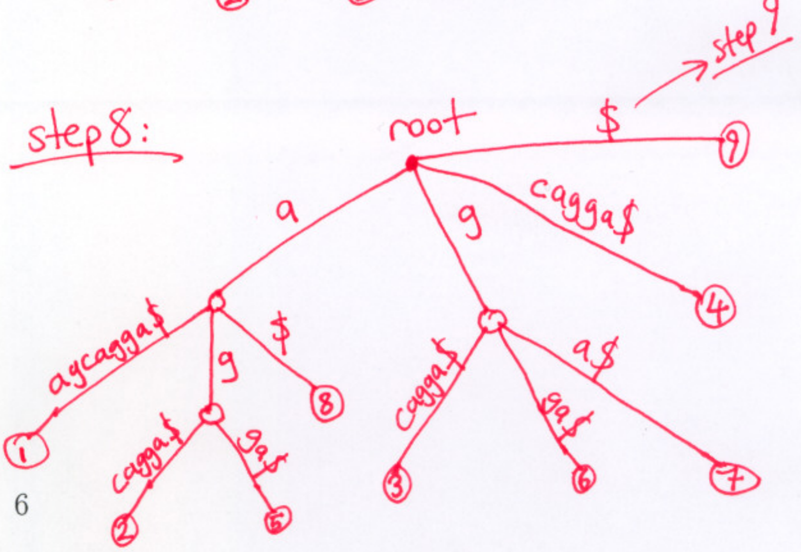
step 6:



step 7:



step 8:



(b)(10 pts) Search the following patterns in the suffix tree you created in part (a). Show your search steps. You may either provide written explanations of the search steps without redrawing the suffix tree you created in part (a), or you may show your steps visually by redrawing the suffix tree. If the pattern is found, indicate the number occurrences of the pattern and show the places of occurrence.

1. (5 pts) **Search pattern: ag**

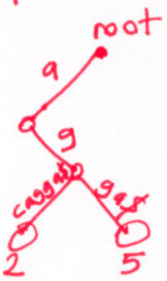
(1) Start from root

(2) First letter "a" matches with the left most branch, proceed with that branch

(3) Second letter "g" matches the right branch of the internal node.

(4) No letter left. Match complete. 2 leaf nodes under the current internal node ⇒ occurs 2 times.

<u>ag</u>cagga$ and <u>ag</u>ga$



2. (5 pts) **Search pattern: gac**

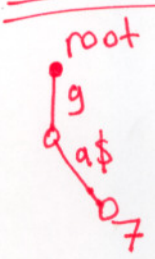
(1) Start from root

(2) First letter "g" matches with the second branch from left, proceed with that branch

(3) Second letter "a" matches with the "a$" branch

(4) Third letter "c" mismatches with "$".

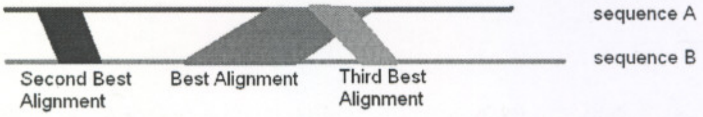
⇒ The pattern <u>does not</u> occur in the string.



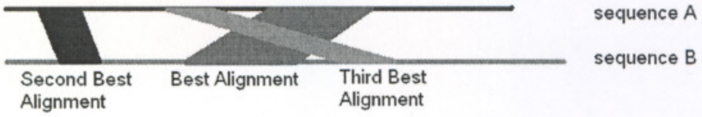
5 **(10 pts)**

Smith-Waterman local alignment technique is used to give the *best* local alignment between two sequences. Suppose, a biologist is also interested in sub-optimal alignments, i.e., second-best, third-best, etc. Describe an algorithm that will find the **best three** ***non-overlapping*** local alignments. **Hint:** The algorithm should be based on analysis of the partial score table constructed by Smith-Waterman method. An example of an ***overlapping*** alignment and an example of a ***non-overlapping*** alignment are given below:

An example of best three ***overlapping*** local alignments (your algorithm **should not** report alignments like these):



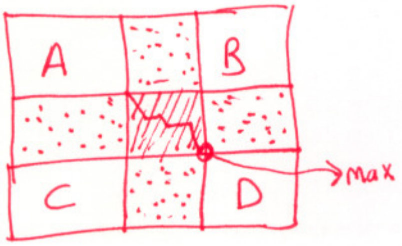
An example of best three ***non-overlapping*** local alignments:



(1) Run Smith-Waterman dynamic programming algorithm and fill in the partial scores table

(2) Find the maximum score in the table and trace back the alignment.

(3) Mark-off the rows and columns overlapping with the alignment (Dotted and shaded regions in the figure below). Marking-off can be achieved by resetting the cells to zero.



(4) Recalculate the partial scores for the regions that are effected by this "marking-off" process. E.g.: For the first iteration recalculate the scores for regions B, C, and D. The score scores in A are the correct scores.

(5) Goto step 2.