

**Bioinformatika**  
**Međuispit**  
 26. studenoga 2018.

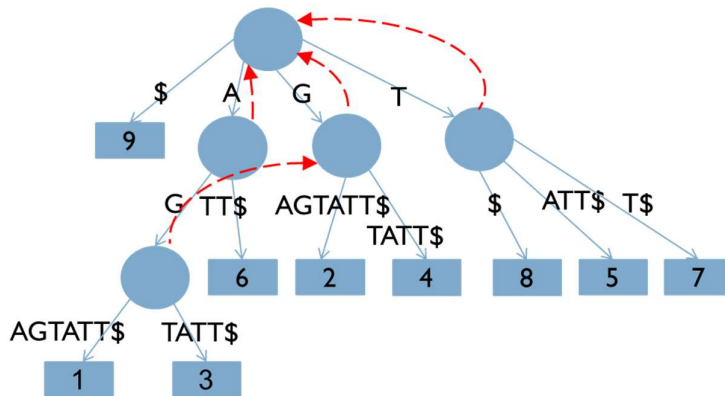
**Rješenja 1.-5. zadatka napisati na vlastitim papirima, a odgovore na 6.-9. pitanje napisati na testu. Sve predati u košuljici.**

**1. (3 boda)**

Za zadani niz  $S = AGAGTATT\$$  potrebno je nacrtati sufiksno stablo. Na sufiksnom stablu označite sufiksne veze. Pretpostavite da je znak  $\$$  leksikografski manji od ostalih znakova niza  $S$ .

**Rješenje:**

1	2	3	4	5	6	7	8	9
A	G	A	G	T	A	T	T	\$



**2. (4 boda)**

Odredite sufiksna polja  $SA$ ,  $SA_1$  i  $SA_2$  te inverzna sufiksna polja  $ISA$ ,  $ISA_1$  i  $ISA_2$  za niz  $S = ACTCGGCG\$$ . Pretpostavite da je znak  $\$$  abecedno manji od ostalih znakova niza  $S$ .

**Rješenje:**

1	2	3	4	5	6	7	8	9
A	C	T	C	G	G	C	G	\$

$SA_1 = [9, 1, 2, 4, 7, 5, 6, 8, 3]$

$SA_2 = [9, 1, 4, 7, 2, 8, 6, 5, 3]$

$SA = [9, 1, 7, 4, 2, 8, 6, 5, 3]$

$ISA_1 = [2, 3, 5, 3, 4, 4, 3, 4, 1]$

ili  $ISA_1 = [2, 3, 9, 3, 6, 6, 3, 6, 1]$

$ISA_2 = [2, 4, 8, 3, 7, 6, 3, 5, 1]$

ili  $ISA_2 = [2, 5, 9, 3, 8, 7, 3, 6, 1]$  ili

$ISA = [2, 5, 9, 4, 8, 7, 3, 6, 1]$

### 3. (4 boda)

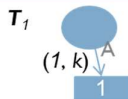
Potrebno je izgraditi sufiksno stablo za niz  $S = \text{ATTA\$}$  korištenjem Ukkonenovog algoritma u vremenu  $O(|S|)$ . Pretpostavite da je znak  $\$$  leksikografski manji od ostalih znakova niza  $S$ .

U svakom koraku izgradnje sufiksnog stabla za niz  $S$  označite koja ste Ukkonenova pravila za dodavanje sufiksa primijenili.

**Rješenje:**

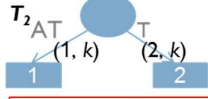
$S = \text{ATTA\$}$

$S[1, 1] = \text{A}$



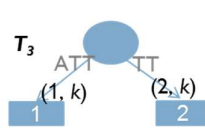
$k = 1$   
 $S[1, 1] \rightarrow \text{pravilo 2}$   
 $j_{\text{pret}} = 1$

$S[1, 2] = \text{AT}$



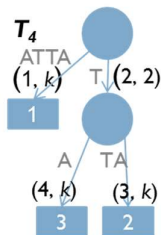
$k = 2$   
 $S[1, 2] \rightarrow \text{pravilo 1}$   
 $S[2, 2] \rightarrow \text{pravilo 2}$   
 $j_{\text{pret}} = 2$

$S[1, 3] = \text{ATT}$



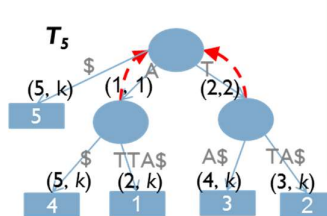
$k = 3$   
 $S[1, 3] \rightarrow \text{pravilo 1}$   
 $S[2, 3] \rightarrow \text{pravilo 1}$   
 $S[3, 3] \rightarrow \text{pravilo 3}$   
 $j_{\text{pret}} = 2$

$S[1, 4] = \text{ATTA}$



$k = 4$   
 $S[1, 4] \rightarrow \text{pravilo 1}$   
 $S[2, 4] \rightarrow \text{pravilo 1}$   
 $S[3, 4] \rightarrow \text{pravilo 2}$   
 $S[4, 4] \rightarrow \text{pravilo 3}$   
 $j_{\text{pret}} = 3$

$S[1, 5] = \text{ATTA\$}$



$k = 5$   
 $S[1, 5] \rightarrow \text{pravilo 1}$   
 $S[2, 5] \rightarrow \text{pravilo 1}$   
 $S[3, 5] \rightarrow \text{pravilo 1}$   
 $S[4, 5] \rightarrow \text{pravilo 2}$   
 $S[5, 5] \rightarrow \text{pravilo 2}$   
 $j_{\text{pret}} = 5, \text{ kraj}$

#### 4. (5 bodova)

Prikažite globalno poravnanje dva niza P = GTCATTGT i S = TAGTCGCA korištenjem Needleman-Wunsch algoritma (umetanje -2, brisanje -2, nepodudaranje -1, podudaranje 2.). Prikažite matricu poravnanja zajedno s početnim uvjetima i samo poravnanje. U matrici prikažite jedan od mogućih puteva poravnanja.

Na temelju prikaza poravnanja odredite kolika bi bila najmanja udaljenost uređivanja ova dva niza.

#### Rješenje:

		G	T	C	A	T	T	G	T
	0	-2	-4	-6	-8	-10	-12	-14	-16
T	-2	-1	0	-2	-4	-6	-8	-10	-12
A	-4	-3	1	-1	0	-2	-4	-6	-8
G	-6	-2	-1	0	-2	-1	-3	-2	-4
T	-8	-4	0	-2	-1	0	1	-1	0
C	-10	-6	-2	2	0	-2	-1	0	-2
G	-12	-8	-4	0	1	-1	-3	1	-1
C	-14	-10	-6	-2	-1	0	-2	-1	0
A	-16	-12	-8	-4	0	-2	-1	-3	-2

#### Poravnanje

GT**C**AT**T**G-**T**

-T**A**GT**C**G**C**A

**Udaljenost uređivanja: 6**

### 5. (4 boda)

Za zadana dva niza P = GTCATTGT i S = TAGTCGCA, popunite tablicu dinamičkog programiranja za računanje polu-globalnog poravnanja (umetanje -2, brisanje -2, nepodudaranje -1, podudaranje 2). U tablici označite put poravnanja početka prvog niza s krajem drugog niza i put poravnanja kraja prvog niza s početkom drugog niza. Ispišite i ta poravnanja.

#### Rješenje

		G	T	C	A	T	T	G	T
	0	0	0	0	0	0	0	0	0
T	0	-1	2	0	-1	2	2	0	2
A	0	-1	-2	1	2	0	1	1	0
G	0	2	0	-1	0	1	-1	3	1
T	0	0	4	2	0	2	3	1	5
C	0	-1	2	6	4	2	1	2	3
G	0	2	0	4	5	3	1	3	1
C	0	0	1	2	3	4	2	1	2
A	0	-1	-1	0	4	2	3	1	0

#### Poravnanje 1 (narančasto):

GTCAT**T**GT-----

-----TAGTCGCA

#### Poravnanje 2 (zeleno):

-----GTCATTGT

TAGTCG-CA

**6. (1 bod)**

Objasnite čemu služe PAM matrice u programu BLAST.

**Rješenje:**

PAM matrice su supstitucijske matrice koje se koriste kod određivanja rezultata poravnanja između sljedova aminokiselina. Svaki element matrice predstavlja stopu mutacije jedne aminokiseline u drugu u nekom vremenskom razdoblju.

**7. (1 bod)**

Objasnite razliku između blastn i blastp programa.

**Rješenje:**

Programom blastn uspoređuju se nukleotidni sljedovi, a programom blastp proteini (sljedovi aminokiselina).

Pretraživanje po nukleotidnim sljedovima se koristi za traženje srodnijih sljedova (manji broj promjena), gdje se očekuju da će riječi (sastavljene od nukleotida) po kojima se pretražuje biti bolje očuvane i dulje, a pretraživanje po sljedovima aminokiselina se koristi za evolucijski udaljenije sljedove, gdje je očuvanost sljedova manja.

**8. (1 bod)**

Objasnite razliku između Hammingove i Levenshteinove udaljenosti dva niza.

**Rješenje:**

Hammingova udaljenost gleda samo zamjene znakova te nizovi moraju biti jednake veličine, dok Levenshteinova udaljenost uzima u obzir zamjene, umetanja i brisanja znakova.

**9. (2 boda)**

Koja je vremenska, a koja memorijska složenost Hirshbergovog algoritma za računanje poravnanja? Objasnite kako je to bolje ili lošije u odnosu na osnovne algoritme dinamičkog programiranja.

**Rješenje:**

Vremenska  $O(N^2)$ , a memorijska  $O(N)$ . Osnovni algoritmi imaju obje složenosti  $O(N^2)$ , što znači da Hirshbergov algoritam ima znatno bolju memorijsku složenost. Iako Hirshbergov algoritam ima jednaku vremensku složenost, biti će otprilike duplo sporiji jer se računati duplo više ćelija matrice (neke će se izračunavati više puta).