

## Uvod u Bioinformatiku, Septembarski rok

Industrija 4.0

Školska godina 2022/2023

### Uputstvo:

Na Desktop-uu se nalazi folder sa nazivom BIO.SEPT u kome se nalaze nepotpuni kodovi rešenja zadataka koje treba dopuniti kako bi traženi algoritmi bili implementirani. Folder preimenovati u format bio\_ispitniRok\_Prezime\_Ime\_BrIndeksa (npr. bio\_sept\_Peric\_Pera\_4005-2021) i sve kodove čuvati u preimenovanom folderu dopunjujući nepotpune kodove.

**NAPOMENA: Kodovi koji se ne kompajliraju (imaju sintaksne greške) automatski nose 0 poena.**

1. [12.5p] Napisati kod *Greedy Motif Search* algoritma koji pronalazi skup motiva dužine  $k$  za dati skup od  $t$  sekvenci `dna_sequences` primenjujući pohlepan pristup.

Primer:

```
dna_sequences = ['ttacggtAAC',
                 'gATAtctgtc',
                 'ACGgcgttcg',
                 'ccctAAAgag',
                 'cgtcAGAggt']
```

$k = 3$

$t = 5$

Izlaz: ['tAA', 'TAt', 'ttc', 'tAA', 'tcA']

2. [12.5p] Napisati kod algoritma *Smith-Waterman* koji pronalazi optimalno lokalno poravnanje između dve sekvence, pri čemu je skor uparivanja istih karaktera jednak 1, skor uparivanja različitih karaktera 0, a poravnanje karaktera sa prazninom -1.

Primer:

```
seq1 = 'OIUGNOISGRVISDIODSHFGMLSVEODRGMSOVFOGHLSDGMOVIDFHGS'
seq2 = 'DSHGMLSEOD'
```

Izlaz: Score: 0

Alignment:

'DSHFGMLSVEOD'

'DSH-GMLS-EOD'

3. [12.5p] Napisati kod *Neighbor-Joining* algoritma koji konstruiše nekoreno evolutivno stablo za matricu rastojanja  $D$  dimenzije  $n \times n$ .

Primer:

```
D = [[0, 13, 21, 22],
      [13, 0, 12, 13],
      [21, 12, 0, 13],
      [22, 13, 13, 0]]
```

$n = 4$

```
Izlaz: { 0: [('3+2+1', 11.0)],
        '3+2+1': [(0, 11.0), ('3+2', 4.0), (1, 2.0)],
        '3+2': [('3+2+1', 4.0), (3, 7.0), (2, 6.0)],
        1: [('3+2+1', 2.0)],
        3: [('3+2', 7.0)],
        2: [('3+2', 6.0)]
      }
```

4. [12.5p] Napisati kod koji korišćenjem strukture *sufiksnog stabla* formiranog za nisku `sequence` pronalazi sva njena uparivanja sa niskom `pattern` i ispisuje sva mesta (indekse) unutar niske `sequence` na kojima je pronađeno uparivanje.

Primer:

```
sequence = 'ATATGCTCTATGCTAGATGTGCTATA'
pattern = 'ATG'
```

Izlaz: [9, 2, 16]