Algorytmy biologii molekularnej
Zadania domowe, seria II
Termin oddania: 12.01.2000

Uwaga: rozwiązanie każdego zadania proszę napisać na osobnej kartce.

Zadanie 1. (30 pkt) Czy prawdziwe jest następujące zdanie:

> Dla dowolnej funkcji podobieństwa s i dla dowolnego optymalnego (w sensie miary SP) uliniowienia $S_1^\#, \ldots, S_k^\#$ dla słów $S_1, \ldots, S_k$, (oraz $k > 2$) istnieje taka para $i \neq j$, że uliniowienie $S_i^\#$ i $S_j^\#$ jest optymalne dla słów S_i oraz S_j.

Uwaga: optymalność jest definiowana jako minimum wartości brane po wszystkich uliniowieniach.

Zadanie 2. (35 pkt) Niech s będzie funkcją podobieństwa spełniającą aksjomaty odległości. W zadaniu tym rozważamy algorytm aproksymacyjny (na wykładzie nazwany uliniowieniem gwiazdowym) służący do uliniowienia k sekwencji długości $O(n)$. Algorytm ten działa w czasie $O(k^2n^2)$. Niech $\overline{sim}(A, B)$ — wartość (minimalizowana) optymalnego uliniowienia dla A i B. Niech $S = \{S_1, ..., S_k\}$ — zbiór sekwencji do uliniowienia. Wyszukujemy jedną z sekwencji $S_i \in S$, (jest to tzw. centrum), dla której $\sum_{j=1}^{k} \overline{sim}(S_i, S_j)$ jest najmniejsza. Następnie dodajemy kolejno wszystkie sekwencje z $S \setminus \{S_i\}$, tak, aby ich uliniowienie z S_i było optymalne (w miarę potrzeby dodając spacje do wcześniej dodanych sekwencji).

(a) Zamiast wybierać S_i w sposób podany powyżej, wybieramy losowe S_i. Wykaż, że nowy algorytm może nie przybliżać optymalnego uliniowienia ze stałym współczynnikiem aproksymacji.

(b) Wykaż, że oczekiwana wartość uliniowienia w algorytmie z punktu (a) jest co najwyżej dwukrotnością wartości optymalnej.

(c) Pokaż, że algorytm z punktu (a) będzie działał w czasie $O(kn^2)$.

(d) Dla dowolnego $r > 2$ i $p < 1$ wskaż randomizowany algorytm aproksymacyjny o czasie działania $O(kn^2)$, przybliżający wartość optymalną ze stałą aproksymacji r z prawdopodobieństwem co najmniej p.

Zadanie 3. (35 pkt) Dla danego uliniowienia $\mathcal{M} = S_1^{\#}, \ldots, S_k^{\#}$ dla słów $S_1, \ldots, S_k$, (oraz $k > 2$) nad alfabetem Σ definiujemy *dominujący znak i-tej kolumny* (może to być spacja) jako znak występujący w tej kolumnie najwięcej razy. Jeśli jest węcej niż jeden taki znak, to dowolny znak występujący maksymalną liczbę razy w danej kolumnie nazwiemy dominującym. Przykład:

```
      R  G  T  T  -  -  Y  A  Q  -  W
      R  D  -  T  -  P  T  E  Q  V  -
      R  D  -  -  A  P  -  G  Q  -  W
---------------------------------------
      R  D  -  T  -  P  Y  G  Q  -  W
```

Dominujące słowo to słowo powstałe przez konkatenację dominujących znaków kolejnych kolumn. Jeśli a jest i-tym znakiem dominującego słowa, definiujemy *niezgodność* dla i-tej kolumny jako liczbę znaków różnych od a w tej kolumnie. Oznaczmy tę liczbę przez $d(i)$. Za *niezgodność uliniowienia* $\mathcal{M}$ przyjmujemy liczbę $\Sigma_{i=1}^{l} d(i)$, gdzie l jest liczbą kolumn $\mathcal{M}$. Problem *optymalnego dominującego uliniowienia* polega na znalezieniu uliniowienia $\mathcal{M}$ o minimalnej niezgodności. Zaprojektuj algorytm aproksymacyjny dla tego problemu.