Algorithms in Bioinformatics

Andrey Prjibelski (andrewprzh@gmail.com),
Ira Vasilinetc (vasilinetc.ira@gmail.com)

March 23, 2014

Task 9

For given tree in linear time find maximal set (of vertices) that does not contain connected vertices.
Input:
Tree in adjacency list format (number of vertices is given before the list).
Output:
Size and elements of the maximal disconnected set.
Example:

Input:
3
1 2
1 3
Output:
2
2 3

Task 10

Let T be a binary rooted tree that contains sequences of equal length in its leaves and internal nodes. We define

$$d_H(T) = \sum_{\{s,t\} \in E(T)} d_H(s,t)$$

where $d_H(s,t)$ is the Hamming distance between s and t, where s and t are the sequences corresponding to the neighboring nodes of the tree.

For a given tree T of some species we know the sequences corresponding to the leaves of T. The goal as to find sequences corresponding to the internal nodes such that $d_H(T)$ is minimal.
Input:
Tree T of n species ($n \leq 500$) in the Newick format and the corresponding multiple alignment of sequences of equal length in FASTA format (length $\leq$ 300 bp) that correspond to the leaves of T.

Output: The minimum possible value of $d_H(T)$, followed by a collection of sequences to be assigned to the internal nodes of T that will minimize $d_H(T)$. If multiple solutions exist, output one.

Example:

Input:
(((ostrich,cat)rat,(duck,fly)mouse)dog,(elephant,pikachu)hamster)robot;
>ostrich
AC
>cat
CA
>duck
T-
>fly
GC
>elephant
-T
>pikachu
AA

Output:
8
>rat
AC
>mouse
TC
>dog
AC
>hamster
AT
>robot
AC

Task 11

Let T be a binary tree of n vertices ($n \leq 100$) with the root t. For each edge there is an integer number representing its weight. For a given integer $q < n$, we remove some edges, so that the remaining graph is also a binary tree of q edges with the root t and the maximal possible total weight.

Input:
First line contains two numbers n and q. The following $n-1$ lines are the adjacency list with weights. Each line contains numbers of ending vertices of the edge and its weight. The root is always numbered as 1.

Output:
Maximal possible total weight of the remaining q edges.

Example:

Input: 5 2

1 3 1
1 4 10
2 3 20
3 5 20
Output:
21

Task 12

Let the tree T of n vertices represent a road map. For each pair of neighboring cities (i, j) the traveling time $t(i, j)$ is given. The goal is to calculate the average traveling time for all possible pairs of cities, i.e.

$$t_{avg} = \frac{\sum_{1 \leq i,j \leq n, i \neq j} t(i, j)}{n(n-1)}$$

Input:
First line contains the number of cities n ($n \leq 50000$). The following $n-1$ lines are the adjacency list with the time for each road.
Output:
Average traveling time for all possible pairs of cities with a precision of 4 decimal digits.
Example:
Input: 4
1 2 1
2 3 1
2 4 1
Output:
1.5000