CSE3800/BME4800/CSE5800: BIOINFORMATICS
Programming Assignment #1

Due Tuesday, Sept. 27, 2011

The *closest sequence problem* is defined as follows:

Given n DNA sequences $S_1,\ldots,S_n$, each of length m, find a DNA sequence t of length m such that $d:=max_{i=1,\ldots,n}\ d_H(t, S_i)$ is minimized, where $d_H(t, S_i)$ denotes the Hamming distance between t and S_i.

For this project you must implement a method for finding optimal solutions to the closest string problem. You can either implement a branch-and-bound algorithm or use integer programming in conjunction with optimization engines such as the *glpsol* solver distributed as part of the GNU Linear Programming Kit (http://www.gnu.org/software/glpk/).

Input

Your program should read from the standard input a line containing integers *n* and *m*, followed by *n* lines each containing a DNA sequence of length *m*.

Sample input

```
10 25
CTGGCGGTGGCTATCATCCGTCCCT
CATGCGAGTGGTCGGTGATAGCTCG
GAAGTGTGAGGAATCCGTAGAGAAT
GAACTAAGTAGTTCACCTTACCCTC
CCAACACTCATATCGTCTTGCTACT
TGACTCCTTTTTTATTCATATTTTC
AATACTCGACCTTCCACGAAGGCTG
GGATTCACCTCCCTTTCCGCTGAAT
CAGAGGTAAAAGAAAGGGGGACAAT
GATAATCGTAGAATTAAATAAGACA
```

Output for branch and bound implementation

If you implement a branch-and-bound algorithm your program should print to the standard output a line containing an optimum sequence *t* followed by a line containing $max_{i=1,\ldots,n}\ d_H(t, S_i)$.

Sample output

```
CAAATGCGTACTATCTCGTATCAAT
d = 15
```

Output for integer programming

If you use integer programming, your program should print to the standard output an integer program model of the input problem instance in lp format (see http://lpsolve.sourceforge.net/5.1/CPLEX-format.htm).

Turn-in instructions

Submit on HuskyCT the following:

- A 1-2 page write-up with a high-level description of your branch-and-bound algorithm or a description of your integer programming formulation (variables used, constraints, etc.)
- Computer code in the programming language of your choice implementing the branch-and-bound algorithm or for generating the integer program models. Include instructions for running the code if using an exotic programming language.
- The solution to at least two of the attached instances, computed either by running your branch-and-bound algorithm or by running an integer programming solver such as *glpsol* on the lp file generated by your code. For Windows users solutions can be obtained by loading the lp model in the freely available *GLPK Lab* (http://glpklabw.sourceforge.net/) which integrates *glpsol*.