Assignment 2

1. Write code to read a DNA data matrix. You can choose between reading a data file that has a very simple format or reading a data file following a simplified version of the NEXUS format.

   (a) If you choose the first option, assume that the first line of the data file contains two numbers, the number of taxa and the number of characters. Then each following line contains a word, which is the name of a taxon, and a line of DNA characters.

   (b) If you choose the second option, you need to be able to read a NEXUS data block with a 'dimensions' and 'format' statement followed by a 'matrix' specification.

   The DNA characters can either be A, C, G, or T. Write the code such that you can also recognize commonly used symbols to denote ambiguity and matches with the state of the first taxon in the matrix (Table 1). Examples of the two data formats are given below.

2. Write code to calculate the length of a given tree using a specified data matrix and Fitch optimization. Also write code to perform the same task using Sankoff optimization under a specified cost matrix. Then write a wrapper class to test your code. Your Main function (in the test class) should ask the user for the name of the data file, the name of the tree file, and the transition and transversion costs to be used in the Sankoff optimization. The transition cost applies to the pyrimidine (C-T) and purine (A-G) changes and the transversion cost to all other changes. Then calculate the length (cost) of the tree under Fitch and Sankoff parsimony and output these values.

3. (Optional) Find the optimal states for all interior nodes in the tree under Fitch and Sankoff parsimony and output those in a suitable format for a character that the user is interested in.

Make sure that your code is well documented and follows good programming practices. Mail the source code and a jar file with your program to Peter or Fredrik no later than **Monday September 19**. Let us know what format your program expects for the tree file and the data file.

An example of the simple matrix format follows below:

```
5 18
taxon_1 ACG GGT TTG CGA GGC GAT
taxon_2 ACG GGT TCG CGG GGT AAC
taxon_3 ACG AGT TCA CGA GAT AAT
```

```
 taxon_4 ACA AGC TCA CGT GAC AAA
 taxon_5 ACA AGC TCA CGT GAC AAT
```

The same matrix would be written like this in our simplified version of the NEXUS format:

```
 #NEXUS

begin data;
    dimensions ntax=5 nchar=18;
    format datatype=DNA matchchar=. gapchar=- missing=?;
    matrix
    taxon_1 ACG GGT TTG CGA GGC GAT
    taxon_2 ACG GGT TCG CGG GGT AAC
    taxon_3 ACG AGT TCA CGA GAT AAT
    taxon_4 ACA AGC TCA CGT GAC AAA
    taxon_5 ACA AGC TCA CGT GAC AAT
    ;
end;
```

In both formats, it is true that white space has no special meaning except as a delimiter outside of the data matrix. So, for instance, there must be at least a space between the taxon name and the DNA characters but there could be two or more spaces as well. In the Nexus format, the options 'ntax', 'nchar', 'datatype', 'matchchar', 'gapchar', and 'missing' can come in any order. Note that the match character is used to mean 'state same as for the first taxon at this position'. Using the match character, the matrix we have described above could be given in the NEXUS format as:

```
 #NEXUS

begin data;
    dimensions ntax=5 nchar=18;
    format datatype=DNA matchchar=. gapchar=- missing=?;
    matrix
 taxon_1 ACG GGT TTG CGA GGC GAT
 taxon_2 ... ... .C. ..G ..T A.C
```

```
 taxon_3 ... A.. .CA ... .AT A..
 taxon_4 ..A A.C .CA ..T .A. A.A
 taxon_5 ..A A.C .CA ..T .A. A..
    ;
end;
```

In the 'format' statement of the NEXUS data block, only the datatype option is required. You can assume that the other options have the default values given in our example file even though, strictly speaking, this is not part of the Nexus standard. The state codes for DNA data, including the standard symbols for missing characters and gaps, are given in Table 1. Note that, typically, both missing characters and gaps are interpreted as complete ambiguity concerning the state at the tip.

Table 1: Codes used for DNA characters. The two last character codes ('-' and '?') are the standard codes for gaps and missing characters, respectively.

Code	State set	Binary repr.
A	{A}	0001
C	{C}	0010
M	{A,C}	0011
G	{G}	0100
R	{A,G}	0101
S	{C,G}	0110
V	{A,C,G}	0111
T	{T}	1000
W	{A,T}	1001
Y	{C,T}	1010
H	{A,C,T}	1011
K	{G,T}	1100
D	{A,G,T}	1101
B	{C,G,T}	1110
N	{A,C,G,T}	1111
-	{A,C,G,T}	1111
?	{A,C,G,T}	1111