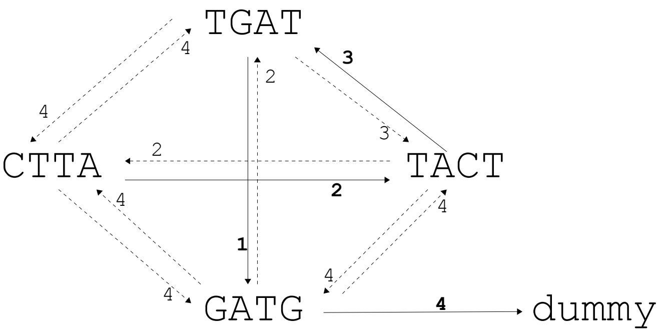
Model Solutions for Exercise 5 (Algorithms for Bioinformatics)

13.10.2011

1. Build the prefix graph, and add dummy node that has an incoming edge from each node v having a weight equal to the length of the string s_v. Solve ATSP for this graph using brute-force (NP hard, works only for really small inputs). Since there are no outgoing edges from the dummy node, it will be the last vertex visited.



(*Note: only one edge leading to dummy node is visible/drawn.*)

The resulting superstring:

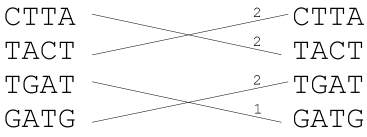
```
CTTACTGATG      (superstring)
CTTA
  TACT
     TGAT
      GATG
```

Solving ATSP will always yield the optimal result (OPT = 10 here).

2. There are two cycles in the minimum weight cycle cover of the prefix graph:

The first cycle, of weight $2+2=4$, is between `CTTA` and `TACT`. The second cycle, of weight $1+2=3$, is between `TGAT` and `GATG`.

The minimum weight perfect matching corresponding to those cycles is:



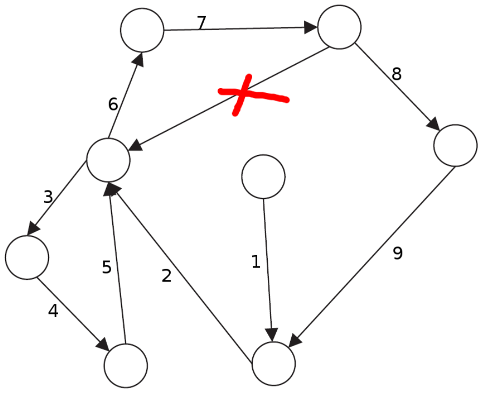
The resulting superstring is (some) concatenation of strings corresponding to those cycles:

```
CTTACTTGATG      (superstring)
CTTA
  TACT
      TGAT
       GATG
```

The approximation factor *for this particular input* is $\frac{11}{10} = 1.1$ (i.e. length of the resulting string divided by OPT).

Instead of concatenation, you can overlap the strings corresponding to cycles. By chance, with this example this gives the optimal solution. One can also show that by overlapping the worst case approximation factor drops from 4 to 3 (see Vazirani, Chapter 7).

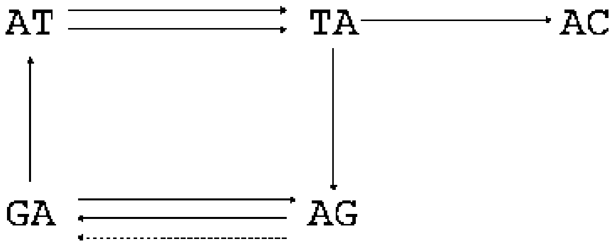
3. Deleting one edge is enough:



4. The multiset of ℓ-mers is $S = \{\texttt{GAG}, \texttt{GAT}, \texttt{TAG}, \texttt{ATA}, \texttt{ATA}, \texttt{AGA}, \texttt{TAC}\}$.

   The set of $\ell - 1$-mers is $S = \{\texttt{GA}, \texttt{AG}, \texttt{AT}, \texttt{TA}, \texttt{AC}\}$.

   Build a graph that has the set of $\ell - 1$-mers as vertices and the set of ℓ-mers as edges:



   (*Note: The edge with dashed line does not belong to the original graph*)

   Notice that the graph does *not* contain an Eulerian path; there must be a missing ℓ-mer. We can find the missing ℓ-mer by adding edge(s) to the graph until the resulting graph contains an Eulerian path: now if we add one edge from AG to GA (the edge with dashed line), the resulting graph contains the Eulerian path AT-TA-AG-GA-AG-GA-AT-TA-AC which corresponds to string ATAGAGATAC. Notice that Spectrum of this string is the multiset $S \cup \texttt{AGA}$ where AGA is the missing 3-mer from the original spectrum.

5. Only showing here the connection of ultrametric tree and the three-point condition version considered at the lecture (two distances are equal, third smaller).

   Fix i and j and let their lowest common ancestor be c. Consider all 4 ways how k could be located with respect to c: (i) in a subtree preceding that of c, (ii) in the left subtree of c (with i), (iii) in the right subtree of c (with j), and (iv) in a subtree succeeding that of c. It is easy to derive that for (i) and (iv) we have $d_{ij} < d_{ik} = d_{jk}$, for (ii) we have $d_{ik} < d_{ij} = d_{jk}$, and for (iii) we have $d_{jk} < d_{ij} = d_{ik}$. Hence, assuming an ultrametric tree, we see that the three-point condition holds for all i, j, k.

   Other direction follows from the correctness of UPGMA algorithm (see study group 5): Take minimum d_{ij} and from three-point condition it follows $d_{ik} = d_{jk}$ for all k. Hence for the next step matrix D can be modified by deleting row j and column j (row i becomes the parent of old i and j). Repeating this process on D one notices that only if the condition does not hold for some i, j, k, the UPGMA algorithm result cannot be interpreted as an ultrametric tree.