Algorithms for Bioinformatics (Autumn 2014)

Exercise 4 (Tue 30.09., 10-12, B222)

1. **Shortest common superstring and ATSP.**

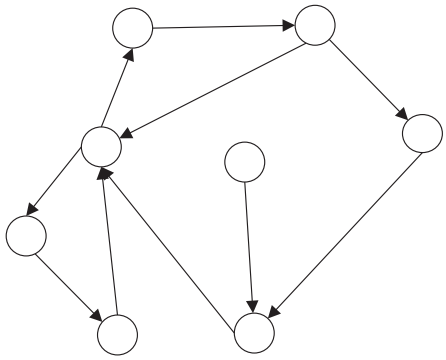
   Solve the shortest common superstring problem on set $S = \{\texttt{CTTA},\texttt{TGAT},\texttt{TACT},\texttt{GATG}\}$ by reducing the problem to asymmetric traveling salesman problem through the prefix graph and dummy vertex as described at the lecture.

2. **Shortest common superstring and minimum weight cycle cover.**

   Simulate the 4-approximation algorithm for shortest common superstring problem on the same set S as above. Visualize also the minimum weight perfect matching corresponding to the minimum weight cycle cover. What is the real approximation factor achieved on this instance?

3. **Graph editing.**

   Eulerian path in a graph is a path that visits all *edges* exactly once. Insert and delete minimum number of edges to/from the graph below so that it has an Eulerian path.



4. **Sequencing by hybridization.**

   A measurement from a hybridization experiment estimates that the 3-mer spectrum of s would be $Spectrum(s,3) = \{\texttt{GAG},\texttt{GAT},\texttt{TAG},\texttt{ATA},\texttt{ATA},\texttt{AGA},\texttt{TAC}\}$. Construct s by the Eulerian path approach described at the course, taking into account that there might be one ℓ-mer missing from the measured spectrum. *Hint: Note that because the spectrum contains twice the 3-mer* `ATA`*, the graph should have two edges from* `AT` *to* `TA`*.*

5. **Preprocessing for gene rearrangement study.**

   Consider you have the genome sequences of two species A and B and you would like to study their rearrangement distance. Each gene in A may have several putative homologs with different local alignment score in B, and vice versa. How would you find a one-to-one mapping between all genes in A to genes in B so that the sum of the corresponding local alignment scores is maximized? Here we may assume that A has at most as many genes as B (otherwise their role can be switched). *Hint. Reduce to a graph problem and add some dummy nodes/edges.*