Advanced Algorithms - Problem set 4

Whenever you give an algorithm, also argue for its running time and correctness. Always feel free to ask on MS Teams if you're stuck, need a clarification, or suspect a typo.

1. Experiments with TSP approximations:

   - Describe a dynamic programming algorithm for finding a TSP in a general graph G. The algorithm should run in $O(2^n p(n))$ time for some polynomial p and output a Hamiltonian cycle of minimum total weight in G. Argue for the correctness of this algorithm and implement it.
   - Implement the 2-approximation algorithm for Metric TSP discussed in the lecture. Your implementation may use inefficient data structures when implementing the MST algorithm.
   - Compare the output of the 2-approximation algorithm and the DP when given as input the list of Danish cities on the next page. (Use the haversine formula or the Spherical Law of Cosines to compute the distances.)
   - For a bonus, plot the two tours from the previous part on a map.

2. Experiments with LPs:

   - Implement the 2-approximation algorithm for Vertex Cover based on *maximal* matchings.
   - Implement the LP-based rounding algorithm for Vertex Cover using calls to an LP solver of your choice.
   - Compare the output of the 2-approximation algorithm and the LP-based algorithm on the following graphs and interpret the results:
     - A random graph with 200 vertices in which each edge is sampled independently with probability 0.9.
     - The attached 200-vertex graph (encoded as a list of edges).
   - If your LP solver outputs an integral solution, what can you conclude?

3. In a Set Cover instance with universe $U = \{x_1, \ldots, x_n\}$ and sets $S_1, \ldots, S_m$, the frequency of element x_i is the number of sets among $S_1, \ldots, S_m$ that contain x_i. Let f be the maximum frequency over all elements in U.

   (a) Give the ILP formulation of Set Cover and describe how to obtain an LP relaxation of this ILP.

   (b) Describe an LP-based rounding algorithm for Set Cover that results in an f-approximation.

Some Danish cities

```
Copenhagen, 55.676 / 12.566
Aarhus, 56.157 / 10.211
Odense, 55.396 / 10.388
Aalborg, 57.048 / 9.919
Esbjerg, 55.47 / 8.452
Horsens, 55.861 / 9.85
Randers, 56.461 / 10.036
Kolding, 55.49 / 9.472
Vejle, 55.709 / 9.536
Greve, 55.583 / 12.3
Svendborg, 55.060337 / 10.611613
Thisted, 56.956957 / 8.686066
Holstebro, 56.358404 / 8.613281
Aabenraa, 55.045335 / 9.419403
Faaborg, 55.098016 / 10.244751
Grenaa, 56.413142 / 10.879211
```