28. Consider the following variation of the TSP: Find the minimum cost/length "tour" where each vertex/city has to be visited at least once (i.e., multiple visits are allowed).

    (a) Prove that for cost matrices that fulfill the triangle inequality there always exists a Hamiltonian cycle (TSP tour) which is optimal (i.e., where each vertex is visited exactly once).

    (b) Prove that if the triangle quality is not fulfilled, the problem above can be solved by solving an appropriately constructed TSP instance with a distance matrix that fulfills the triangle inequality.

29. Apply the following heuristics/approximation algorithms to the TSP instance with the distance matrix D

$$D = \begin{pmatrix} - & 2 & 1 & 3 & 6 & 3 \\ 2 & - & 3 & 4 & 8 & 5 \\ 1 & 3 & - & 4 & 6 & 2 \\ 3 & 4 & 4 & - & 4 & 5 \\ 6 & 8 & 6 & 4 & - & 8 \\ 3 & 5 & 2 & 5 & 8 & - \end{pmatrix}$$

    (a) double tree algorithm

    (b) Christofides algorithm.

30. Consider the following instance of the metric TSP: We are given the points $P_1 = (1,1)$, $P_2 = (4,2)$, $P_3 = (3,6)$, $P_4 = (9,3)$, $P_5 = (4,8)$, $P_6 = (12,5)$, $P_7 = (11,1)$. The distances are given by the Manhattan distance between the points.

    (a) Apply the MST/double tree heuristic to obtain a tour.

    (b) Apply the Christofides heuristic to obtain a tour.

    (c) Are the tours obtained in (a) and (b) optimal for the 2-opt neighbourhood?

31. Modify the Christofides algorithm such that it can be applied to get a $\frac{3}{2}$-approximation for the traveling salesman path problem (instead of a Hamiltonian cycle one searches for a Hamiltonian path) in the metric case (distance matrix is symmetric and fulfills the triangle inequality).

32. Consider the Euclidean TSP (all vertices are embedded into the $\mathbb{R}^2$ and the edge lengths are equal to the Euclidean distance of the vertices they connect).

    (a) Prove that in an optimal solution of the problem no two edges cross each other.

    (b) Find a class of instances for which the quality of the double tree algorithm is arbitrarily close to the upper bound 2.

    (c) Find a class of instances for which the quality of the Christofides algorithm is arbitrarily close to the upper bound $\frac{3}{2}$.

33. Consider the special case of the symmetric TSP where all distances are 1 or 2.

    (a) Show that the 2-opt heuristic is a constant factor approximation algorithm for this case. Provide a tight performance bound.

    (b) (More challenging) Try to improve on (a) by using a different algorithm.

34. Apply the 2-opt heuristic with initial tour $\phi = (1,2,3,4,5)$ to the TSP with distance matrix D

$$D = \begin{pmatrix} - & 3 & 4 & 1 & 2 \\ 3 & - & 2 & 5 & 3 \\ 4 & 2 & - & 6 & 2 \\ 1 & 5 & 6 & - & 7 \\ 2 & 3 & 2 & 7 & - \end{pmatrix}$$

35. Consider the Euclidean TSP in the plane. Prove or disprove the following statement: The 2-opt procedure delivers a crossing-free tour.

36. How many possibilities exist to create a new tour from a 3-opt step for the symmetric TSP?

37. Devise an exact algorithm for the TSP which is based on dynamic programming. What is the running time of your approach? (Hint: Consider variables $h(x, R)$ with $R \subseteq \{2, \dots, n\}$, $x \in R$. Let $h(x, R)$ be the cost of the shortest Hamiltonian path from 1 to x which visits all vertices from $R \setminus \{x\}$ as intermediary vertices.

38. Apply the algorithm developped in Problem 37 to find an optimal tour for the TSP instance with distance matrix

$$D = \begin{pmatrix} 0 & 2 & 9 & 10 \\ 1 & 0 & 6 & 4 \\ 15 & 7 & 0 & 8 \\ 6 & 3 & 12 & 0 \end{pmatrix}$$

39. Apply the construction heuristics (nearest neighbour and other insertion heuristics) introduced in the lecture to the following TSP instance

$$D = \begin{pmatrix} - & 4 & 3 & 4 & 4 & 6 \\ 4 & - & 2 & 6 & 3 & 5 \\ 3 & 2 & - & 4 & 1 & 3 \\ 4 & 6 & 4 & - & 5 & 4 \\ 4 & 3 & 1 & 5 & - & 2 \\ 6 & 5 & 3 & 4 & 2 & - \end{pmatrix}$$

40. Suppose the n cities of a TSP instance are partitioned into m clusters such that the distance between two cities is zero if and only if they belong to the same cluster.

    (a) Prove that there exists an optimal tour with at most $m(m-1)$ edges of positive weight.

    (b) Prove that this special case of the TSP can be solved in polynomial time if m is fixed.

41. Show that the nearest neighbour heuristics can perform arbitrarily bad for instances of the TSP with a distance matrix which does not fulfill the triangle inequality.

42. Prove that it suffices to require the 2-matching inequalities for sets F, $F \subset \delta(X)$, $|F|$ odd where F is a matching. The remaining inequalities are implied.

43. Provide a direct proof that every TSP tour fulfills the 2-matching inequalities.

44. Show that the 2-matching inequalities are a special case of the comb inequalities.

45. Show that the subtour inequalities, the 2-matching inequalities and the comb inequalities are special cases of the clique tree inequalities.

46. (METRIC BIPARTITE TSP) Let G be a complete bipartite graph $G = (V, E)$ with bipartition $V = A \cup B$ where $|A| = |B|$. Further let $c : E \mapsto \mathbb{R}_+$ a cost function with $c((a, b)) + c(b, a')) + c((a', b')) \geq c((a, b'))$ for all $a, a' \in A$, $b, b' \in B$. The task is to find a Hamiltonian cycle in G with minimum cost.

    (a) Prove that the existence of a k-approximation algorithm for the METRIC BIPARTITE TSP implies the existence of a k-approximation algorithm for the METRIC TSP.

    (b) Find a 2-approximation algorithm for the METRIC BIPARTITE TSP.

47. Transform the so-called multiple travelling salesman problem (mTSP) to a classical TSP. In the mTSP we are given n cities 0, …, n-1, an $n \times n$ distance matrix D and m salesmen which all start from and turn back to a depot (city 0). The task is finding (sub)tours for the m salesmen so that every city except the depot is visited exactly once (by one of the salesman) and so that the sum of the costs of the m (sub)tours is minimized.

48. Model the following variation of the mTSP (see Problem 47). In addition to the input of the mTSP we are given a fixed charge d_j for every salesman j, $j = 1, \ldots, m$. In this problem variant not all salesmen need to be used - if salesman j is hired, a cost of d_j has to be paid. The task is now to decide which of the salesmen should be hired and then to plan for each of them a subtour such that the overall costs (=sum of the traveled distances plus the sum of the hiring costs) is minimized.

49. Provide an instance of the TSP for which the Held Karp lower bound does not match the cost/length of an optimal tour.

50. Prove that there exist instances of the metric TSP on a complete graph for which the Held Karp bound lies arbitrarily close to $\frac{3}{4}$ times the length of an optimal tour.

51. Let $G = (V, E)$ be a graph with edge costs d_e and let $T = (V, E_T)$ be a minimum spanning tree.
    (a) Show that if vertex 1 is a leaf of T, then an optimum 1-tree can be obtained by adding to the edge set E_T the edge which joins 1 to its second nearest neighbour.
    (b) Provide an example (G, d, T) where the best choice for the vertex 1 (the one providing the largest 1-tree bound) is not a leaf of T.

52. Solve the TSP instance with the distance matrix D given below by applying a branch&bound approach based on the 1 tree relaxation

$$D = \begin{pmatrix} - & 4 & 2 & 4 & 5 \\ 4 & - & 6 & 6 & 8 \\ 2 & 6 & - & 6 & 5 \\ 4 & 6 & 6 & - & 8 \\ 5 & 8 & 5 & 8 & - \end{pmatrix}$$

53. Solve the TSP instance with the distance matrix D given below by applying a branch&bound approach based on the linear assignment relaxation

$$D = \begin{pmatrix} - & 1 & 2 & 3 & 4 \\ 3 & - & 9 & 2 & 1 \\ 7 & 8 & - & 1 & 2 \\ 4 & 3 & 1 & - & 1 \\ 8 & 1 & 1 & 9 & - \end{pmatrix}$$

54. Solve the TSP instance with the following distance matrix by exploiting its structure

$$D = \begin{pmatrix} 4 & 16 & 20 & 23 & 40 \\ 3 & 12 & 15 & 17 & 33 \\ 3 & 10 & 13 & 15 & 31 \\ 1 & 4 & 5 & 6 & 18 \\ 0 & 3 & 3 & 4 & 14 \end{pmatrix}.$$

55. Solve the TSP with the cost matrix D given below (exploit the special structure)

$$D = \begin{pmatrix} 0 & -1 & 7 & -20 & 3 & -2 & 5 \\ 0 & 0 & 12 & 8 & 16 & 9 & 8 \\ 0 & 0 & 0 & 3 & 7 & 6 & 2 \\ 0 & 0 & 0 & 0 & 4 & 4 & 9 \\ 0 & 0 & 0 & 0 & 0 & -18 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

56. A symmetric $n \times n$ matrix D is called Supnick matrix if $d_{ij} + d_{j+1,l} \leq d_{i,j+1} + d_{jl} \leq d_{il} + d_{j,j+1}$ holds for all $1 < i < j < j+1 < l$. Prove that the tour $\langle 1, 3, 5, \ldots, n, \ldots 6, 4, 2 \rangle$ is optimal for the TSP with a distance matrix which is a Supnick matrix.

57. Let D be a nonnegative $n \times n$ matrix which is graded up its columns, i.e. if $d_{ij} \leq d_{i+1,j}$ holds for all i, j. Let φ be an optimal solution for the linear assignment problem with cost matrix D. Prove that one can construct from φ a TSP tour τ such that $d(\tau) \leq d(\varphi) + \max_{i,j \in \{1,\ldots,n\}} d_{ij}$.

58. Prove that the TSP for doubly graded matrices (all row vectors and all column vectors are either monotone increasing or monotone decreasing) is not easier to solve than the general TSP.

59. The bottleneck TSP (BTSP) results from the classical TSP by changing the way the cost of a tour is obtained: instead of summing up the costs of all tour edges one takes the maximum of the costs of the tour edges. An $n \times n$ matrix D is called a large matrix if $d_{ij} = \max\{u_i, v_j\}$ for all i, j with $u, v \in \mathbb{R}^n$. Show that the BTSP for large matrices can be solved in polynomial time.

60. Consider the cycle shrinking algorithm described below. Show that it is an $O(\log n)$ approximation algorithm for the asymmetric TSP with nonnegative distances that fulfill the triangle inequaliy.

    - For a graph with a single vertex output the cycle consisting of the single vertex.
    - Otherwise, find a minimum cost cycle cover (set of cycles covering all vertices, i.e. a subgraph in which each of the vertices has in-degree 1 and out-degree 1) with cycles $C_1, \ldots, C_k$ where each C_i contains at least 2 vertices.
    - From each cycle C_k pick an arbitary proxy vertex v_k.
    - Solve the problem recursively on the graph induced by the vertex set $\{v_1, \ldots, v_k\}$ to obtain a solution C.
    - Determine an Eulerian cycle C' for the graph induced by the edge set $C \cup C_1 \cup \cdots \cup C_k$. Shortcut C' to obtain a Hamiltonian cycle for the original graph and output it.