# Hand-in exercise 2 in
# Linear and Combinatorial Optimization, 2018.

Work in pairs, and complete exercises 1, 2, and 3 below. Hand in one solution per pair of students by sending (i) the matlab file for the functions boundy.m and branchy.m and (ii) the results for exercise 2 and 3 to me (sara@maths.lth.se). Remember to write the name of both students on everything you submit.

*Due date: 13 February 2018*

1. The following two functions solve the travelling salesman problem given a distance matrix $D$, using the branch and bound method. *Download them from the course homepage.*

   ```
   function [x,fopt]=travsalesman(D);
   function [x,fopt]=branchandbound(x,D,minmax,fopt);
   ```

   They use two functions branchy.m and boundy.m to solve the problem. Construct the two MATLAB-functions boundy.m and branchy.m that have the following form:

   ```
   function bounds=boundy(x,D,minmax);
   % function bounds=boundy(x,D,minmax);
   % calculates the 1x2 vector with lower and upper bound
   % respectively,
   % given the 1xn vector with the current path,
   % the NxN distance matrix D and the Nx2 matrix
   % minmax, where minmax(i,1) is the minimum distance
   % from city i and minmax(i,2) is the maximum distance
   % from city i.

   function X=branchy(x,N);
   % function X=branchy(x,N);
   % returns the mx(n+1) matrix X where
   % each row of X is a possible extension
   % of the input path x.
   % x is a 1xn vector, and N is the total number
   % of cities in the problem.
   ```

2. Download the script *travsalesmandata.m* (containing a matrix $D$) from the course homepage, and run the function travsalesman.m on the matrix $D$.

3. Construct a number of problems with different number of cities, and give a plot of the execution time as a function of number of cities.