# Clustering Methods

Exercises 2/7, 25.1.2022

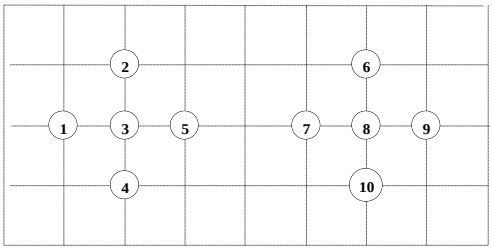
**Five tasks required. If done more, each earn one bonus point.**

1. Extend your clustering tool by implementing (a) nearest neighbor search, (b) optimal partition. The nearest neighbor search has two inputs: data point and the data where the nearest is searched from. It can be either the original dataset or set of centroids. Optimal partition has two inputs: dataset and set of centroids. It loops over all data points and finds the nearest centroid for each of them. Output is a partition of the data (cluster labels). Generate optimal partition for your centroids last week exercise #1 and calculate its *sse*-value. Verify its correctness using *Clusterator*.

2. Implement (a) centroid function that takes a set of data points as input and outputs their average by looping each dimension independently and constructing centroid as the averages of all dimensions. Implement (b) centroid step for k-means where the input is data and a partition. Can you now implement k-means algorithm? If yes, do so. Run it and calculate *sse* of the final result.

3. In your k-means algorithm implement a method to keep track on the *activity* of the centroids. Run k-means algorithm and output the following statistics after each iteration: (a) sse-value, (b) number of active centroids (both absolute and percentage). Collect the results in a table and/or plot as a graph. Use your own data or *any* data from the *Clustering basic benchmark* in http://cs.uef.fi/sipu/datasets/. It is encouraged to select dataset different than others.

4. Assume that the x-axis is used instead of the diagonal for the MPS project with the data below. Demonstrate how the method works when our input is the middle point between 4 and 10, and the points are processed from 1-10 in this order.




5. Assuming that each K-means iteration takes O(*Nk*)-time. Assume also that each random swap takes O($\alpha N$), and that two iterations of the same k-means is applied after every swap. Assuming that k-means requires 25 iterations to converge, how many trial swaps can we perform using the same time if $\alpha$=4?

**Bonus tasks:**

6. Implement random swap operation to your software. Apply k-means for 5 iterations. Then apply 100 different random swaps to the same k-means result. Calculate the *sse* before and after the swaps. Count how many times *sse* improves. Explain the result.

7. Using the clustering animator (http://cs.uef.fi/sipu/clustering/animator/), estimate the probability $p$ that k-means will find the correct clustering for set $S_2$. Repeated k-means (RKM) applies k-means several ($R$) times starting from a different random initialization. Using your estimate $p$, calculate the probability that the correct clustering is found in $R$ iterations. Assuming that K-means is applied 25 iterations until convergence, on average, plot your probability estimation (y-axis) as a function of k-means iterations (x-axis).

8. The figure below shows one k-means initialization technique. Explain how it works and why it does not work?