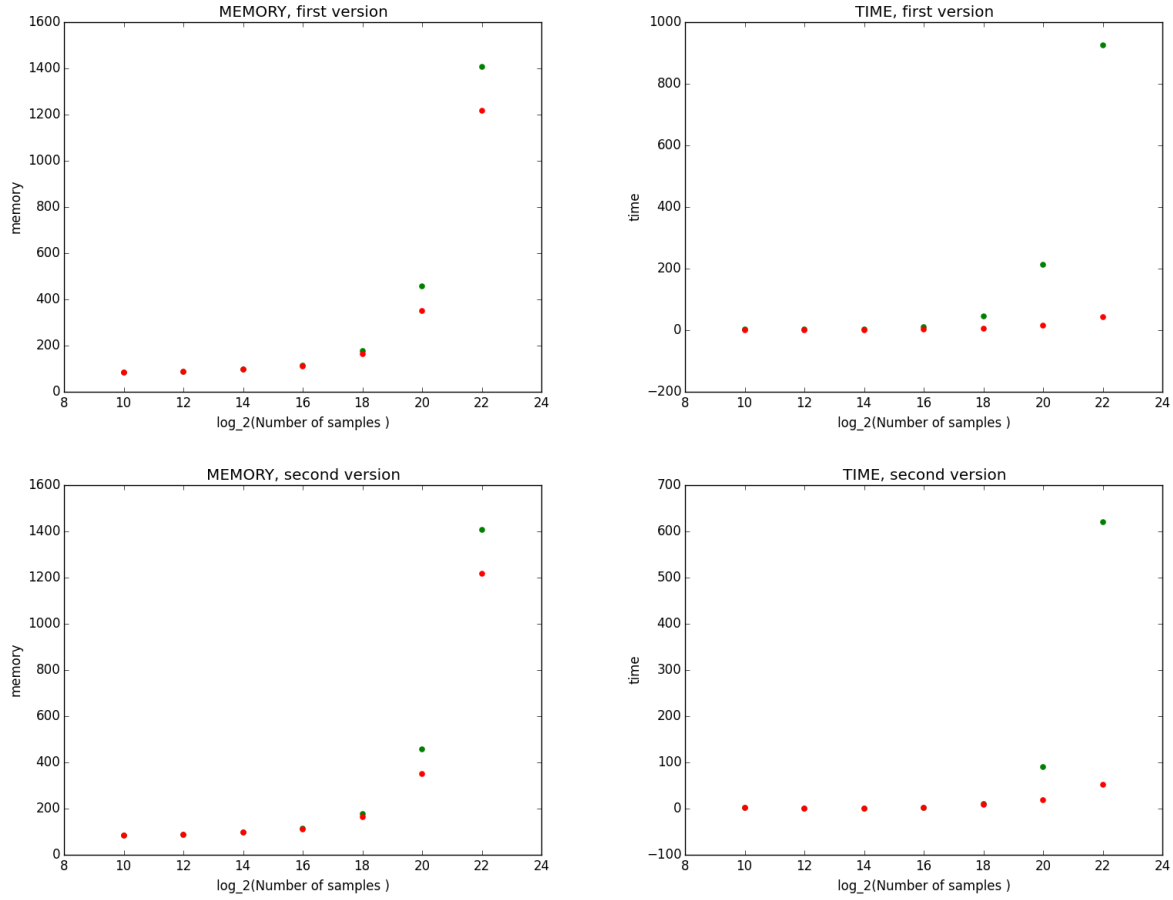
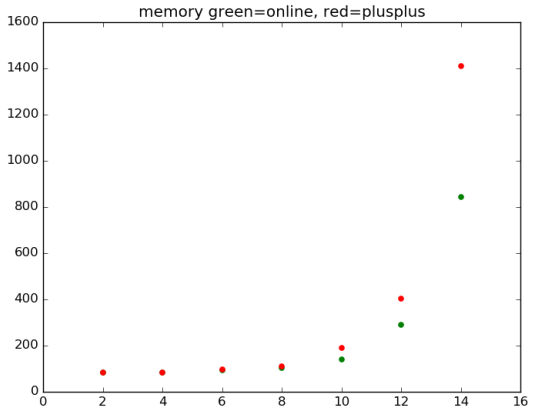
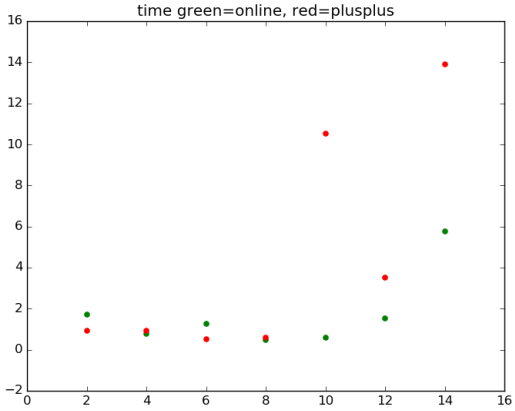


1 Online Kmeans

In order to test the performance of our implementation of the kmeans algorithm we have ran some series of tests using NYU HPC. We have created a synthetic data set consisting of a serie of clusters with gaussian distribution. For each run we have fixed the number of clusters and we have increased the number of points, so to evaluate how the performance changes. We used a space of 32 dimension, as the dimension of our project. Compareing our version of online kmeans and the one in scikit-learn we observe an exponential increase of the resources (memory and time) while the number of samples increase. In the plot our version of kmeans is plotted in green and the scikit-learn in red. The number of samples is represented using logarithm in base

2.
In order to improve our version we implemented an exit strategy inspired to the scikit code. In each iteration of the algorithm we evaluate the inertia of the cluster (sum of the distances off all the points from the closer centroid) and we break out of the for loop when there is not any improvement on the inertia for 10 cycles in a row. With this strategy we improve a little the time exponential behaviour (see plots first and second version).





2 Kmeans plus plus

We have implemented the kmeans++ with the same exit strategy we used for the online kmeans and than compared the two version we had. In the plot, online kmeans (in green) perform better than kmeans++ (in red) for a big number of samples and for big number of cluster.
From this test we conclude that online kmeans is better for a set of sample greater than 2^{10} points