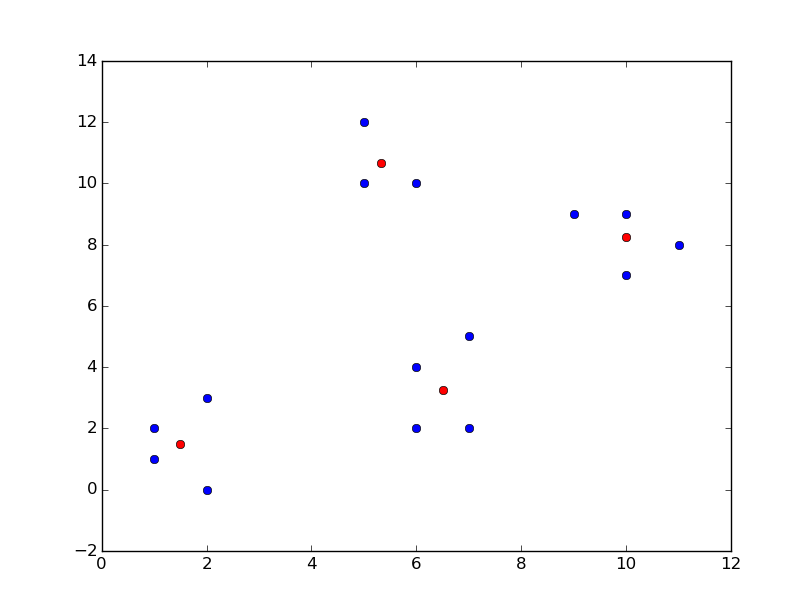
# Part B. Scalable Clustering

1. Practice the k-means algorithm using Euclidean distance.
2. The new clusters (i.e., the examples belonging to each cluster)
   1. Data points in cluster 1 is: A1(1, 1), A2(1,2), A3(2, 0), A4(2, 3)
   2. Data points in cluster 2 is: A5(5, 10), A6(5, 12), A9(6, 10)
   3. Data points in cluster 3 is: A7(6, 2), A8(6, 4), A10(7, 2), A11(7, 5)
   4. Data points in cluster 4 is: A12(9, 9), A13(10, 7), A14(10, 9), A15(11, 8)
3. The centers of the new clusters

The centers of the new cluster is (1.5, 1.5), (), (6.5, 3.25), (10, 8.25).

To illustrate it, we plot the centers in red dot with the original data points (blue dots)



1. How many more iterations are needed to converge? Please write down the final centers.

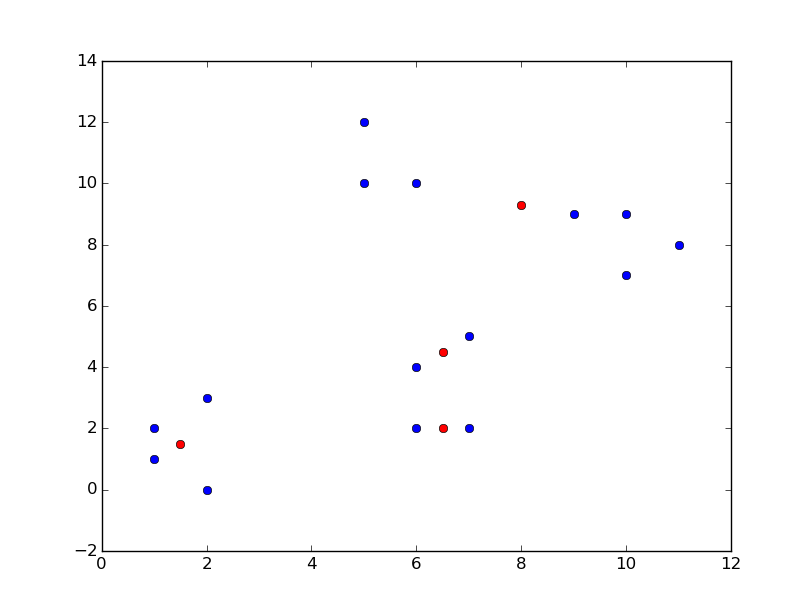
After the first iteration, we need **one** more iteration to converge (0 if you do not count converge check as a whole iteration).

The final centers is still (1.5, 1.5), (), (6.5, 3.25), (10, 8.25).

1. If we use A7, A8, A10, A11 as initial seeds, how many more iterations are needed to converge? Please write down the final centers.

If we choose (A7, A8, A10, A11) as the initial centers, then **we need 3 iterations (2 if you do not count converge check as a whole iteration) in total**.

From following figure, we can see that, if we choose (A7, A8, A10, A11), we will get different clustering result. The final centers are (1.5, 1.5), (6.5, 4.5), (6.5, 2.0), (8.0, )



1. How to accelerate the k-means algorithm using MapReduce? Please specify how you design the mapper and reducer, as well as the whole structure.

Mapper:

* Read data points from the standard input (file on the disk).
* For each data point we compute its distance to the centers. Assign this point to the nearest center.
* Emit (, p), here is the nearest center and p is the data point.

Reducer:

* For data points in the the same cluster, we compute the average of the coordinates and **update** the coordinate of the centers.
* If the new centers are close to old centers and the summed distance is less than the specified distance, we terminate the map reducer process. Else we turn to mapper.

1. Given the same 15 data points in the previous question. And we still cluster them into four groups with the same initial seeds (A1, A5, A7, A12). Please answer the following questions:
   1. If you use the BFR algorithm, please present the results as (N, SUM, SUMSQ).
2. Cluster 1(N, SUM, SUMSQ) = (4, (6,6), (10, 14))
3. Cluster 2(N, SUM, SUMSQ) = (3, (16, 32), (86, 344))
4. Cluster 3(N, SUM, SUMSQ) = (4, (26, 13), (170, 49))
5. Cluster 4 (N, SUM, SUMSQ) = (4, (40, 33), (402, 275))
   1. What is the advantage of the representation of (N, SUM, SUMSQ) compared to the k-mean algorithm?
6. **Overcome the bottleneck of memory limit.** BFR reads data from disk one-main-memory-full at a time. It is easy to update the mean and standard deviation for each cluster while not keeping all the data in memory. BFR is more efficient for large data set because it only scans the data for once.
7. **Accelerate computation.** In the BFR algorithm, the representation (N, SUM, SUMSQ) is used, while in the k-means algorithm, the centroid is used. The advantage of using (N, SUM, SUMSQ) instead of centroid and the standard deviation in each dimension for the BFR algorithm is that this representation avoids unnecessary re-computation during update. If we use centroid instead, then each update involves an additional multiplication by N. The update of the standard deviation is even more complex. Using (N, SUM, SUMSQ) enables a more efficient update as we only need to add the new value directly, without involving N in the calculation.