Matteo Manzini - 2140689
Jacopo Onorati - 2154067

Analysis of G16HW1.py

Description of MRPrintStatistics()

Some general notations:

L = number of partitions; K = set of centroids; N = number of points; k_i = centroid i in K.

Round 1

- Map phase: for every 0 <= j < N and 0 <= i < |K| separately ($coord_j$, g_j) -> (k_i, ($coord_j$, g_j))

  where $coord_j$ are coordinates of point in RDD, g_j is the group of the point of index j.

- Reduce phase: for every 0 <= p < L separately (k_i, ($coord_j$, g_j)) -> (j mod L, (k_{ip}, (SA_{ip}, SB_{ip})))

  where SA_{ip} and SB_{ip} are respectively the number of points with label A and B and related to k_{ip}.

Round 2

- Map phase: -
- Reduce phase: (k_i, S_i) -> (k_i, (NA_i, NB_i)) where S_i is a list of pairs of values SA_{ip}, SB_{ip} related to k_i.

  $NA_i = \sum_{p=0}^{L-1} SA_{ip}$ and $NB_i = \sum_{p=0}^{L-1} SB_{ip}$.

Analysis of MRPrintStatistics()

Rounds = 2

Round 1

- Map phase : O(K)
  In the first map phase we have used flatMap().
  parse_coors_group() is a conversion of points with complexity O(1).
  calc_center() computes the distance from each point to all centroids. The complexity is O(K).
- Reduce phase: O(N*K/L)
  The complexity of mapPartitions() is O(N/L).
  For each centroid, we compute the sum of points A , B in the partition and in the worst case the complexity is O(K).

Round 2

- Map phase: O(1)
- Reduce phase: O(L)
  In input of the last reduce phase we can have at most L tuples, one for each partition.

M_L = O(max{O(K), O(N*K/L), O(1), O(L)}) = O(N*K/L)