# Midterm Review Sheet

Monday, March 2, 2020 1:13 PM

## HAC Clustering:

- Distance between two clusters c1, c2, is determined by pair-wise comparison Of all points in c1 vs all points in c2
  - Single-Link Distance: the minimum distance between some point in c1 and some point in c2
  - Complete-Link Distance: the maximum distance between some point in c1 and some point in c2
  - Mean-Link Distance: computes an artificial "center" point for each cluster and returns the distance between both artificial centers
- Algorithm:
  - While # of clusters is above desired threshold or clusters are within some distance threshold of each other:
    - S1. Make every point its own cluster (n points = n clusters)
    - S2. Find the two closest clusters using the chosen measure of distance given above
    - S3. Merge the two closest clusters found in S2
- Algo-Efficiency
  - There are n-1 rounds because only one cluster is eliminated at each round [O(n)]
  - For each round, there are an average of n comparisons to be made when determining distances [O(n)]
  - For each comparison, we maintain the 'min' with priority queue [O(logn)}
  - [O(n^2 logn)]

## Streaming:

- Approximates the frequency of elements in the stream when the size of the stream prohibits loading the data into memory to get exact counts
- Algorithm(s):
  - **Misra-Greis (Given k and some stream E):**
    - Initialize k-1 labels as null/none and k-1 counters as 0
    - For each element e in stream E:
      - If e matches a label in L
        - Increment the label's count
      - Else if e doesn't match any label
        - Decrement the count of all labels
      - If any label has a count <= 0
        - Replace that label with e
        - Set that labels counter to 1
    - Return all labels L and all counts C
  - **Count-Min Sketch (Given k, j, and some stream E):**
    - Insert Step (builds count-matrix data structure)
      - Initialize a (j)x(k) 0-matrix named CM
      - For each element e in stream E:
        - For each hash function (j of them):
          - Let k_index = hash(e) mod k
          - Increment the count at CM[j][k_index]
    - Query Step
      - Checks each count that e hashes to (j possibilities, one for each hash function), and returns the minimum observed count

## AB Clustering:

- Finds a set of k centers which minimize one of the chosen distance measures below:
  - K-means minimizes the sum of squared distances between some point and its assigned center.
  - K-center minimizes the maximum distance from some point to its assigned center
  - K-median minimizes the sum of distances from some point to its assigned center
  - K-mediod is like k-median except the centers must be actual points in X.
- Algorithm(s):
  - **Gonzales Algorithm (k-center):**
    - S1. Choose some random first center c1 arbitrarily
    - S2. Assign all points to their closest center
    - S3. Find the point farthest from its assigned center
    - S4. Make this point the next center
      S5. Repeat S2-S4 until k centers are chosen
  - **Loyd's Algorithm (k-means):**
    - S1. choose K points (usually from another algo, but can be randomly chosen).
    - S2. Map each point to its closest center c
    - S3. For each c, calculate an artificial "mean" point across all points assigned to c
    - S4. Re-assign each center to its artificial "mean"
    - S5. Repeat S2-S5 until the clusters remain unchanged for newly calculated centers.