Divisive clustering, also known as **top-down hierarchical clustering**, is the opposite of agglomerative clustering. Here, we start with **all data points in a single cluster** and then repeatedly split them into smaller clusters based on their distances until each data point is in its own cluster, or until we reach a desired number of clusters. This approach is less common due to its complexity but is useful when you want to start with broad categories and gradually refine them.

Let's go through the steps of **Divisive Hierarchical Clustering** with a similar example as before.

**Problem Setup**

Suppose we have the same five data points in a 2D space:

* A (1, 1)
* B (1, 4)
* C (5, 1)
* D (5, 4)
* E (3, 3)

**Steps in Divisive Hierarchical Clustering**

1. **Start with All Data Points in a Single Cluster**
   * We begin by treating all data points as part of a single cluster.
   * Initial Cluster: {ABCDE}
2. **Identify the Best Split Based on Distance**
   * Choose a method to measure the distance between points or subsets of points in the cluster.
   * For simplicity, we’ll use **Euclidean distance** and aim to divide points so that each resulting subset has minimal internal distance.
   * For example, let’s try splitting {ABCDE} into two groups by finding two points that are farthest apart and dividing accordingly.
3. **Make the First Split**
   * The two points farthest apart are **A (1, 1)** and **D (5, 4)**, with a Euclidean distance of **5**.
   * We split the cluster by separating points closest to A and D:
     + Cluster 1 (closer to A): {A, B}
     + Cluster 2 (closer to D): {C, D, E}
   * Updated Clusters: {AB}, {CDE}
4. **Continue Splitting Clusters**
   * **Split Cluster {AB}**:
     + Points A and B are relatively close, with a distance of **3.0**, so {AB} might not be further split if stopping at a reasonable number of clusters.
   * **Split Cluster {CDE}**:
     + Next, find the farthest points within {CDE}.
     + Points **C (5, 1)** and **E (3, 3)** have the largest distance within this subset, so we split based on these distances:
       - Cluster 1 (closer to C): {C}
       - Cluster 2 (closer to E): {D, E}
     + Updated Clusters: {AB}, {C}, {DE}
5. **Repeat Until All Points Are Separate Clusters (or Desired Number of Clusters is Reached)**
   * **Final Splits**:
     + If desired, {DE} could be split into {D} and {E}.
     + This yields the final clusters: {A}, {B}, {C}, {D}, {E}.

**Visualizing with a Dendrogram**

In Divisive Clustering, the dendrogram would start with a single cluster at the top and gradually split down into individual points at the bottom. The height at which each split occurs represents the distance between the clusters or subsets being divided.

**Example Applications of Divisive Clustering**

1. **Document or Text Classification**: Useful for dividing large document sets into categories before further refining them into topics or subtopics.
2. **Market Segmentation**: Start with a broad category of all customers, then split based on major differences (e.g., demographic data) and continue refining based on purchasing behavior or preferences.
3. **Taxonomy Creation in Biology**: Begin with a broad classification (e.g., all animals) and gradually divide into species, genus, and family.

**Advantages and Limitations of Divisive Clustering**

**Advantages**

* **Broad to Fine-Grained Approach**: Useful for identifying major groups before diving into finer distinctions.
* **Visual Representation**: The top-down dendrogram provides a clear hierarchical breakdown from broad to narrow groups.

**Limitations**

* **Computationally Intensive**: Divisive clustering is generally more computationally expensive than agglomerative clustering because each split requires assessing all potential splits in a large cluster.
* **Less Common in Practice**: Divisive methods are less commonly implemented in clustering libraries due to their computational cost and complexity.

Divisive clustering offers a complementary approach to agglomerative clustering, providing a top-down view that’s beneficial when initial broad categories are needed before further exploration.