# Description of the algorithm:

**What we have**

* For each **genre**, a ranked list of candidate **videos** with a **score** (higher = better).

**What we want**

* For each genre, pick **exactly k unique videos** (no video can belong to two genres).
* If a video is “fought over” by multiple genres, decide fairly who keeps it.

**Key ideas**

* **Assigned set**: all videos already given to some genre.
* **Fallback option** for a genre: the score of that genre’s **next best not-yet-used** video (the one that would be picked if it loses a conflict). If there’s no next video left, its fallback is considered **very bad**.

**Algorithm steps:**

1. **Initialize videos for each genre**
   * For every genre that has at least **k** candidates:
     + Give it its **top k** videos (highest scores).
     + Remember where you stopped in its list (so you can look at the “next best” later).
   * Ignore genres with fewer than **k** candidates.
2. **Find conflicts**
   * Look for any video that has been assigned to **more than one** genre.
   * If there are no conflicts, you’re done. Go to Step 5.
3. **Resolve each conflict (one contested video at a time)**  
   For the contested video:
   * For each genre that wants this video, compute its **fallback score** (the score of its next best unused video).
   * **Winner rule**: the genre with the **smaller fallback score** keeps the contested video.
     + intuition: the one with the worse alternative needs it more.
   * **Tie-breakers** (only if fallbacks are equal):
     + Prefer the genre that currently has the **higher score on the contested video**.
     + If still tied, the video gets assigned to the genre, which genre name comes first alphabetically.
4. **Refill the losers**
   * Every genre that **lost** the contested video:
     + Remove that video from its list of picks.
     + Take the **next video** that is **not already assigned** to any genre.
     + Add it to this genre’s picks.
     + Keep going until the genre has **k videos** again or runs out of candidates.
     + Move the pointer to the last checked position.
   * After handling all current conflicts, **go back to Step 2** (new conflicts may have appeared).
5. **Finalize**
   * Drop any genre that still ended up with **fewer than k** videos.
   * The remaining genres each have **k unique videos**.

**Output**

* **Assignments**: for each kept genre, the list of its **k** videos (with their scores).