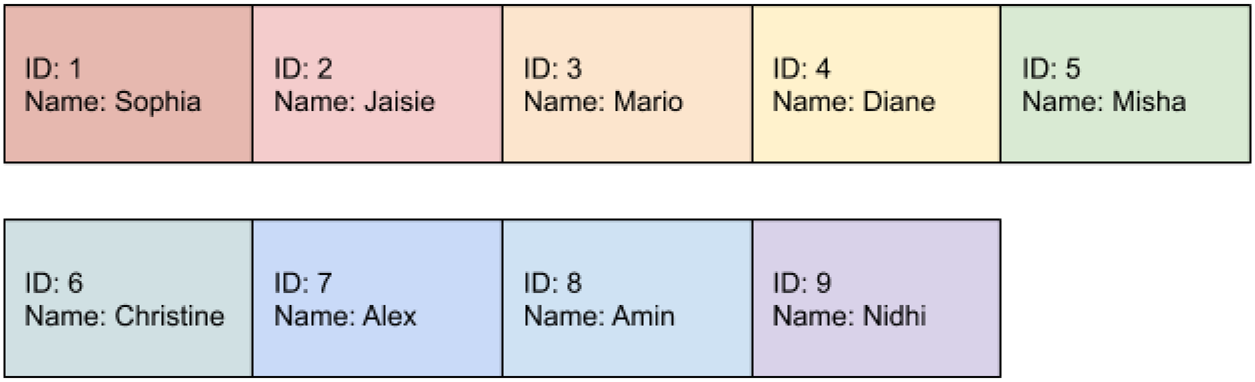
CSC148: A Closer Look at the Grouping Algorithms for Assignment 1

Suppose we have the following set of students in a course. For simplicity, we're only listing their IDs and names:



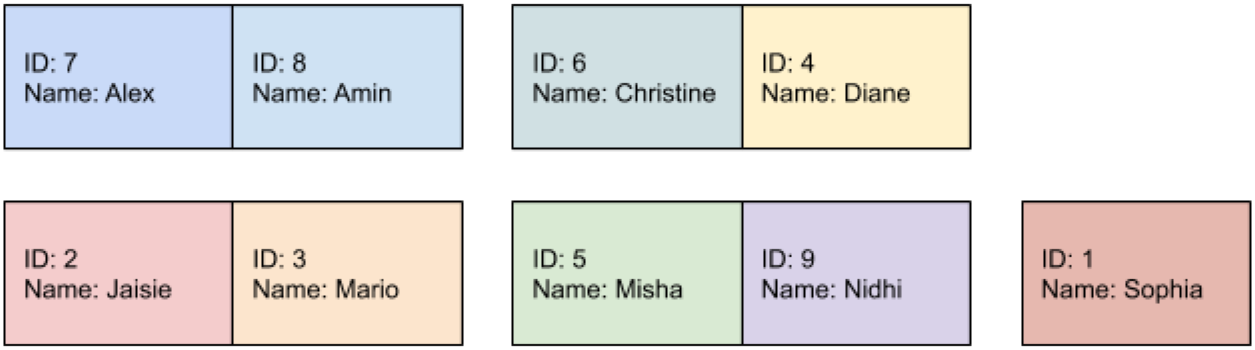
ID: 1 Name: Sophia	ID: 2 Name: Jaisie	ID: 3 Name: Mario	ID: 4 Name: Diane	ID: 5 Name: Misha

ID: 6 Name: Christine	ID: 7 Name: Alex	ID: 8 Name: Amin	ID: 9 Name: Nidhi

Each of the Grouper classes will group these students in different ways, described in their docstrings and below.

AlphaGrouper

The AlphaGrouper class will group students alphabetically. Suppose we're forming groups of size 2. The groups formed should then be:



ID: 7 Name: Alex	ID: 8 Name: Amin

ID: 6 Name: Christine	ID: 4 Name: Diane

ID: 2 Name: Jaisie	ID: 3 Name: Mario

ID: 5 Name: Misha	ID: 9 Name: Nidhi

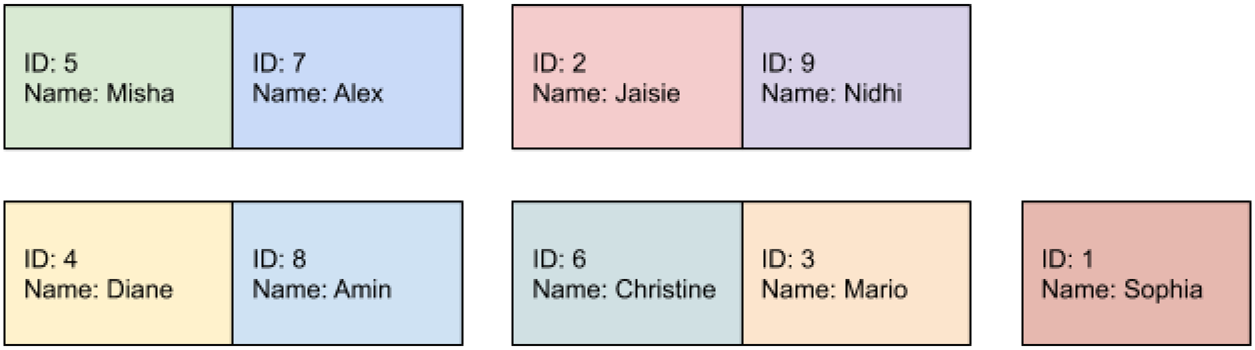
ID: 1 Name: Sophia

Where the students are arranged alphabetically, and then split into groups accordingly.

RandomGrouper

The RandomGrouper class uses no particular ordering to determine groups: students are simply randomly placed into a group. At most, one group can have fewer than the required number of members.

For example, when group sizes are 2, one possible grouping could be:



ID: 5 Name: Misha	ID: 7 Name: Alex

ID: 2 Name: Jaisie	ID: 9 Name: Nidhi

ID: 4 Name: Diane	ID: 8 Name: Amin

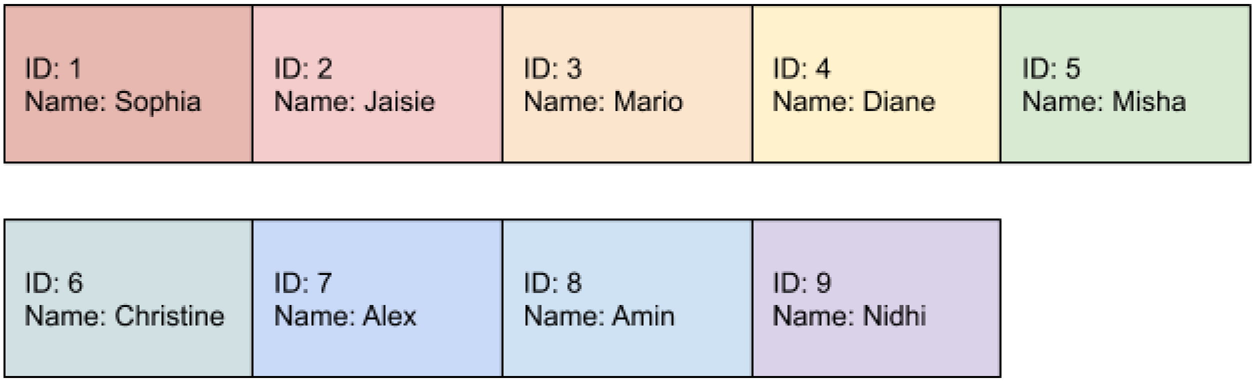
ID: 6 Name: Christine	ID: 3 Name: Mario

ID: 1 Name: Sophia

However, there are many different possibilities, and the groupings can change every time a RandomGrouper makes groups.

GreedyGrouper

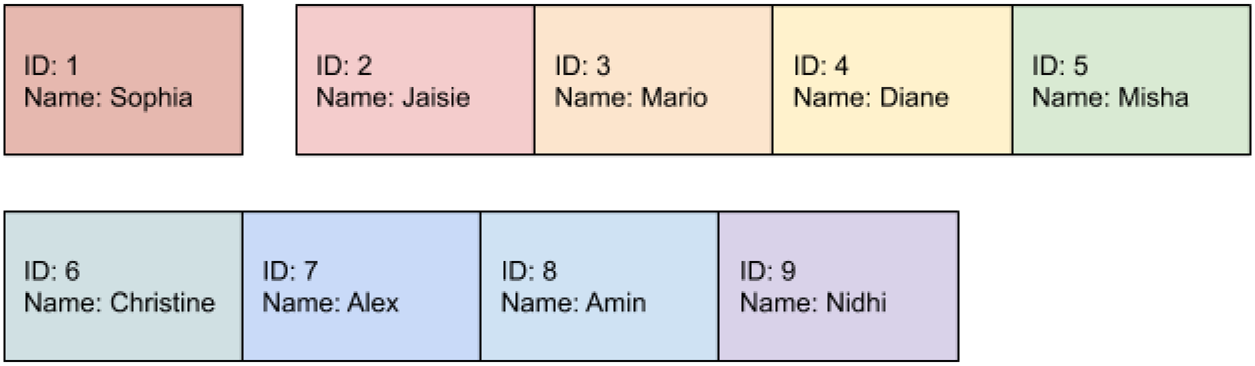
The GreedyGrouper class forms groups in a "greedy" manner: we first get a list of students ordered by ID.



ID: 1 Name: Sophia	ID: 2 Name: Jaisie	ID: 3 Name: Mario	ID: 4 Name: Diane	ID: 5 Name: Misha

ID: 6 Name: Christine	ID: 7 Name: Alex	ID: 8 Name: Amin	ID: 9 Name: Nidhi

The first student who is not part of a group is placed into the first group. In this case, Sophia is in the first group.



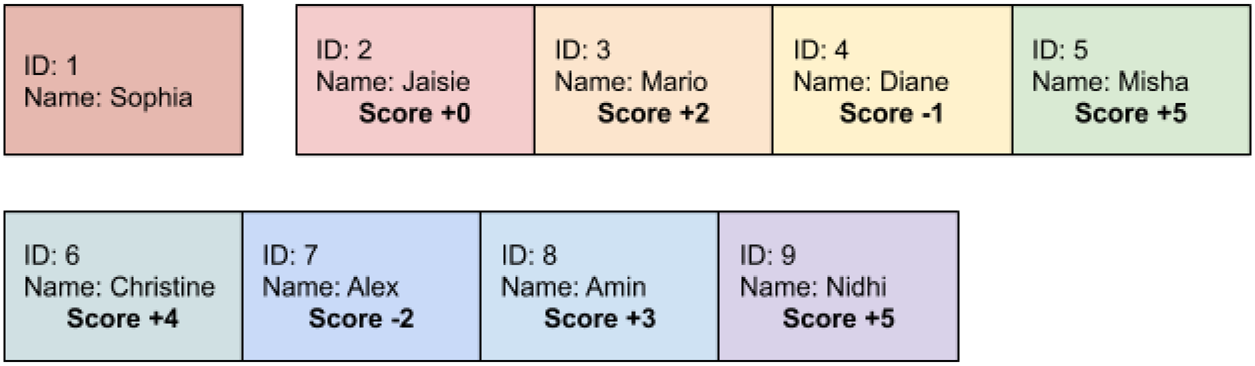
ID: 1 Name: Sophia

ID: 2 Name: Jaisie	ID: 3 Name: Mario	ID: 4 Name: Diane	ID: 5 Name: Misha

ID: 6 Name: Christine	ID: 7 Name: Alex	ID: 8 Name: Amin	ID: 9 Name: Nidhi

Afterwards, the student that would increase this group's score the most (or reduce it the least) will be added to that group, breaking ties by ID.

Suppose we get the following changes to the first group's score for each student:

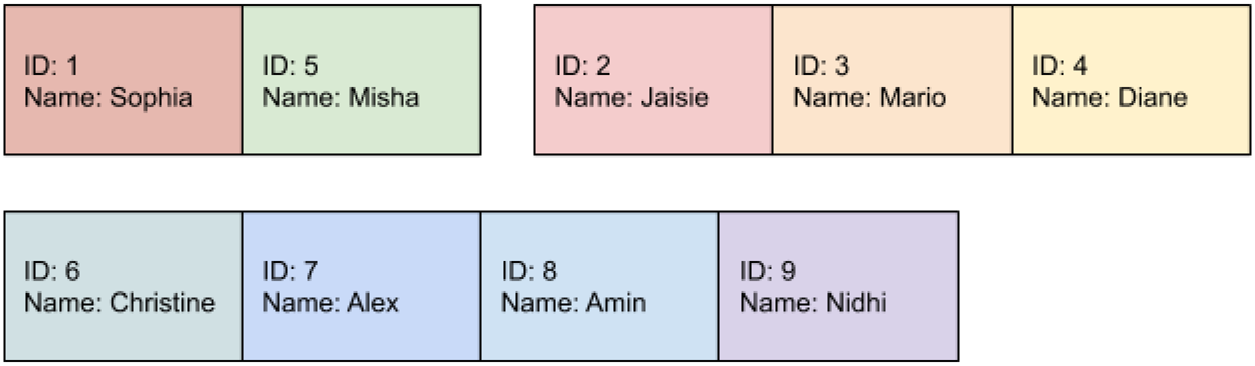


ID: 1 Name: Sophia

ID: 2 Name: Jaisie **Score +0**	ID: 3 Name: Mario **Score +2**	ID: 4 Name: Diane **Score -1**	ID: 5 Name: Misha **Score +5**

ID: 6 Name: Christine **Score +4**	ID: 7 Name: Alex **Score -2**	ID: 8 Name: Amin **Score +3**	ID: 9 Name: Nidhi **Score +5**

Then Misha would be added to the first group:



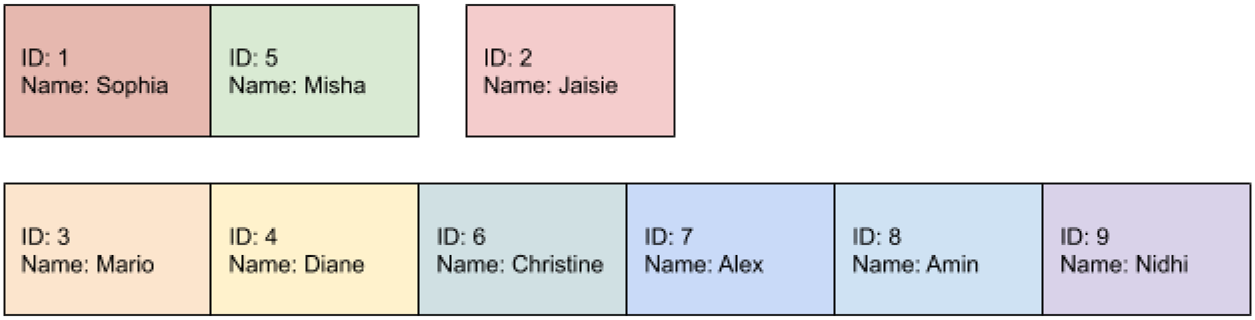
ID: 1 Name: Sophia	ID: 5 Name: Misha

ID: 2 Name: Jaisie	ID: 3 Name: Mario	ID: 4 Name: Diane

ID: 6 Name: Christine	ID: 7 Name: Alex	ID: 8 Name: Amin	ID: 9 Name: Nidhi

We would repeat the previous step until the group reaches our intended size, finding the change in scores for the remaining students and adding the student that would increase the score the most (or decrease it the least) into the group.

Suppose we're using groups of size 2. Then this first group is finished, and we repeat our steps with the next student who has yet to be put into a group, Jaisie:



ID: 1 Name: Sophia	ID: 5 Name: Misha

ID: 2 Name: Jaisie

ID: 3 Name: Mario	ID: 4 Name: Diane	ID: 6 Name: Christine	ID: 7 Name: Alex	ID: 8 Name: Amin	ID: 9 Name: Nidhi

And suppose each student would result in the following changes to the score of Jaisie's group when added:

ID: 1 Name: Sophia	ID: 5 Name: Misha

ID: 2 Name: Jaisie

ID: 3 Name: Mario **Score -15**	ID: 4 Name: Diane **Score -3**	ID: 6 Name: Christine **Score -6**	ID: 7 Name: Alex **Score -3**	ID: 8 Name: Amin **Score -4**	ID: 9 Name: Nidhi **Score -1**

Then Nidhi would be added into Jaisie's group, being the one who reduces it the least:

ID: 1 Name: Sophia	ID: 5 Name: Misha

ID: 2 Name: Jaisie	ID: 9 Name: Nidhi

ID: 3 Name: Mario	ID: 4 Name: Diane	ID: 6 Name: Christine	ID: 7 Name: Alex	ID: 8 Name: Amin

These steps would repeat until we form enough groups.

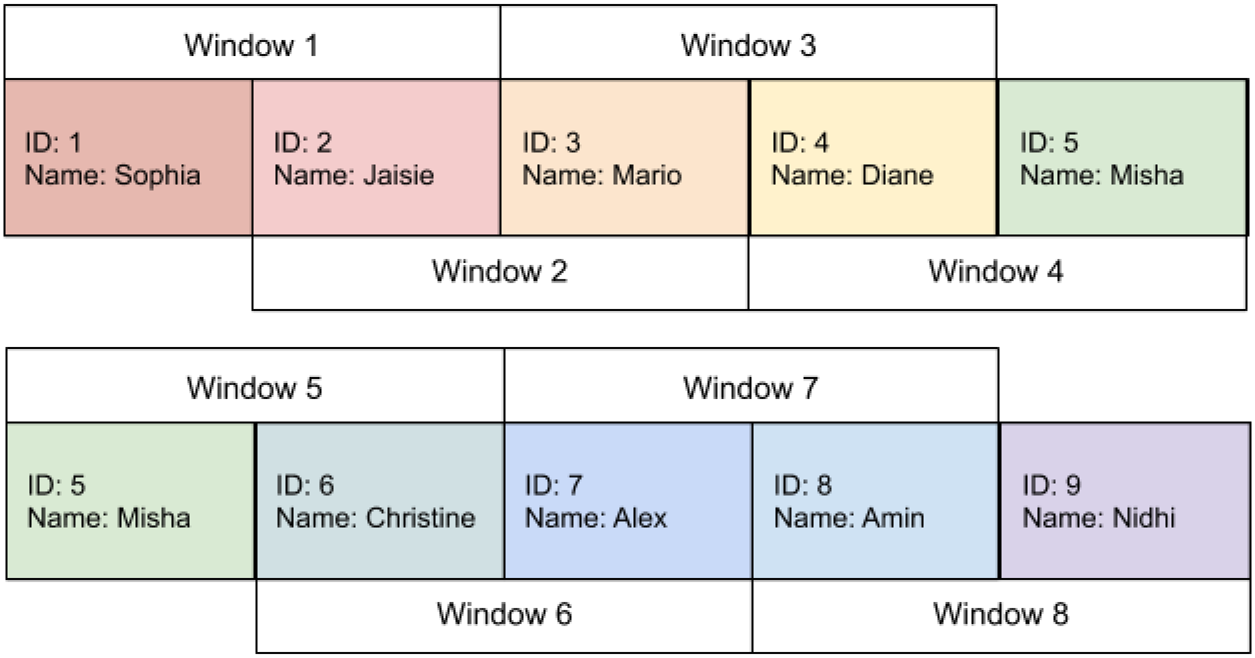
Window Grouper

The WindowGrouper uses 'windows' to form groups. Initially, it starts with a list of students ordered by ID:

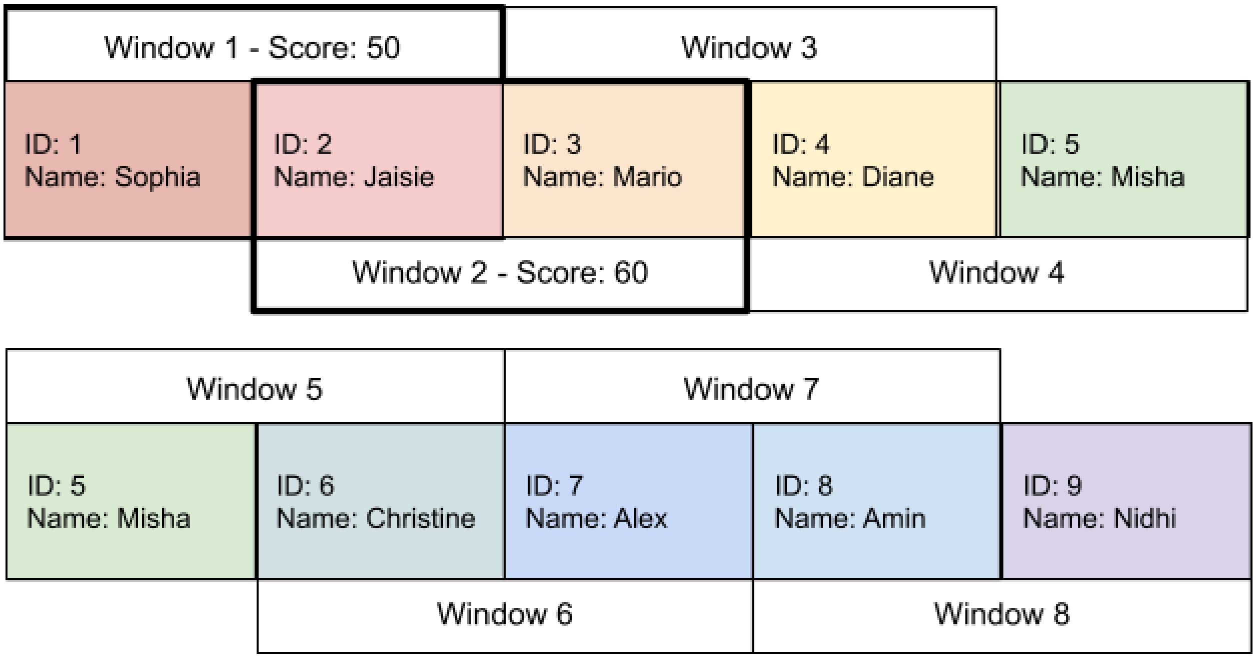
ID: 1 Name: Sophia	ID: 2 Name: Jaisie	ID: 3 Name: Mario	ID: 4 Name: Diane	ID: 5 Name: Misha

ID: 6 Name: Christine	ID: 7 Name: Alex	ID: 8 Name: Amin	ID: 9 Name: Nidhi

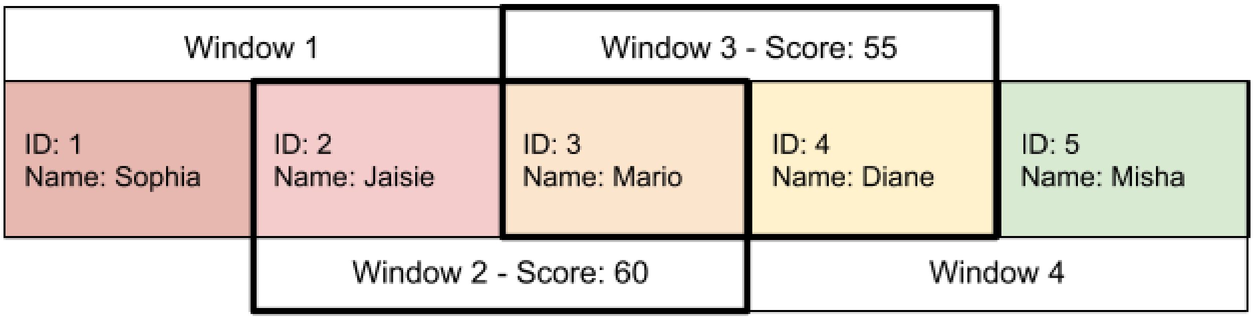
We then get windows for each of the ungrouped students. Suppose we want groups of 2: in this case, our windows would be the following:



We then calculate the score of the first window and the window after it.



If that window's score is greater than or equal to the next window's score, a group is made out of that window. In this case, the score of the next window is higher, so we don't make a group. Instead, we compare the score of window 2 to window 3:



Window 1

Window 3 - Score: 55

ID: 1
Name: Sophia

ID: 2
Name: Jaisie

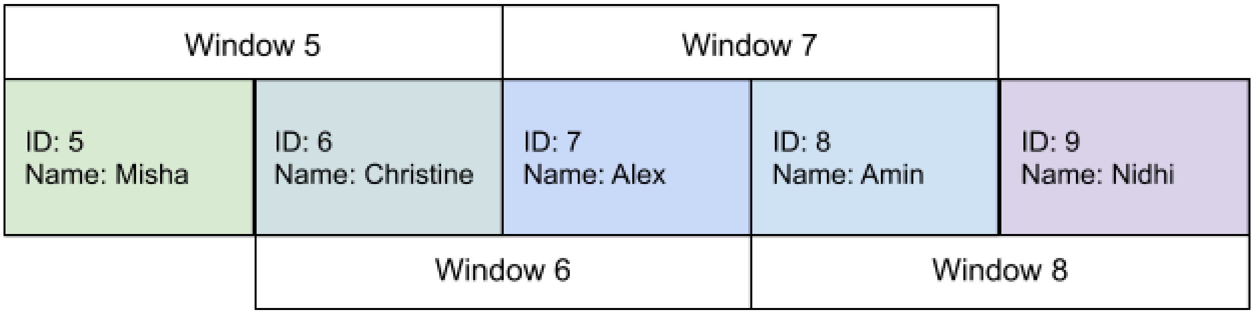
ID: 3
Name: Mario

ID: 4
Name: Diane

ID: 5
Name: Misha

Window 2 - Score: 60

Window 4



Window 5

Window 7

ID: 5
Name: Misha

ID: 6
Name: Christine

ID: 7
Name: Alex

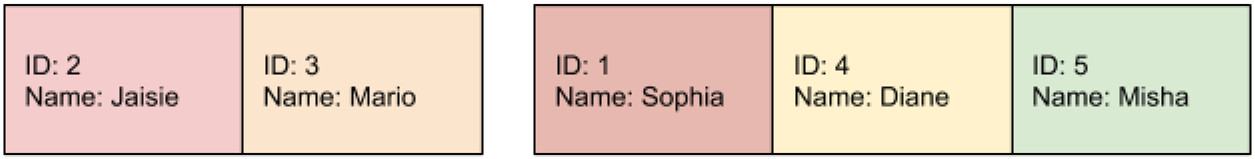
ID: 8
Name: Amin

ID: 9
Name: Nidhi

Window 6

Window 8

In this case, window 2's score is higher than the one after it, so we turn the students in that window into a group, and the remaining 7 students still need to be grouped:



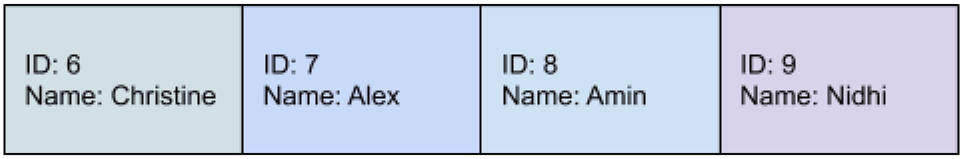
ID: 2
Name: Jaisie

ID: 3
Name: Mario

ID: 1
Name: Sophia

ID: 4
Name: Diane

ID: 5
Name: Misha



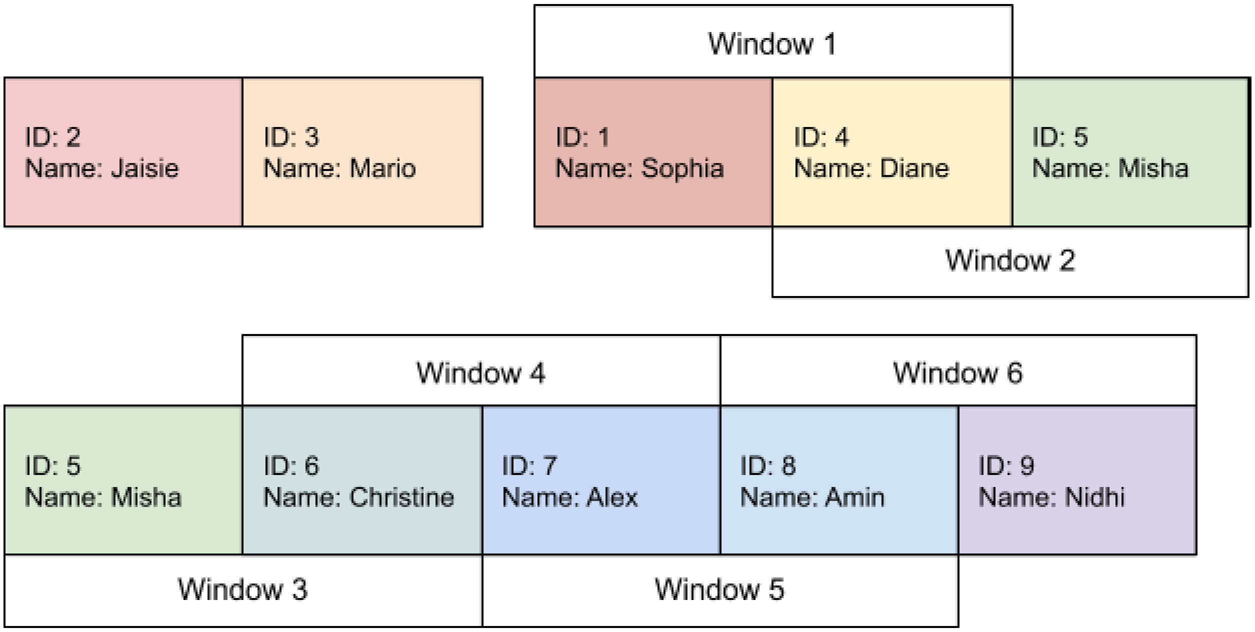
ID: 6
Name: Christine

ID: 7
Name: Alex

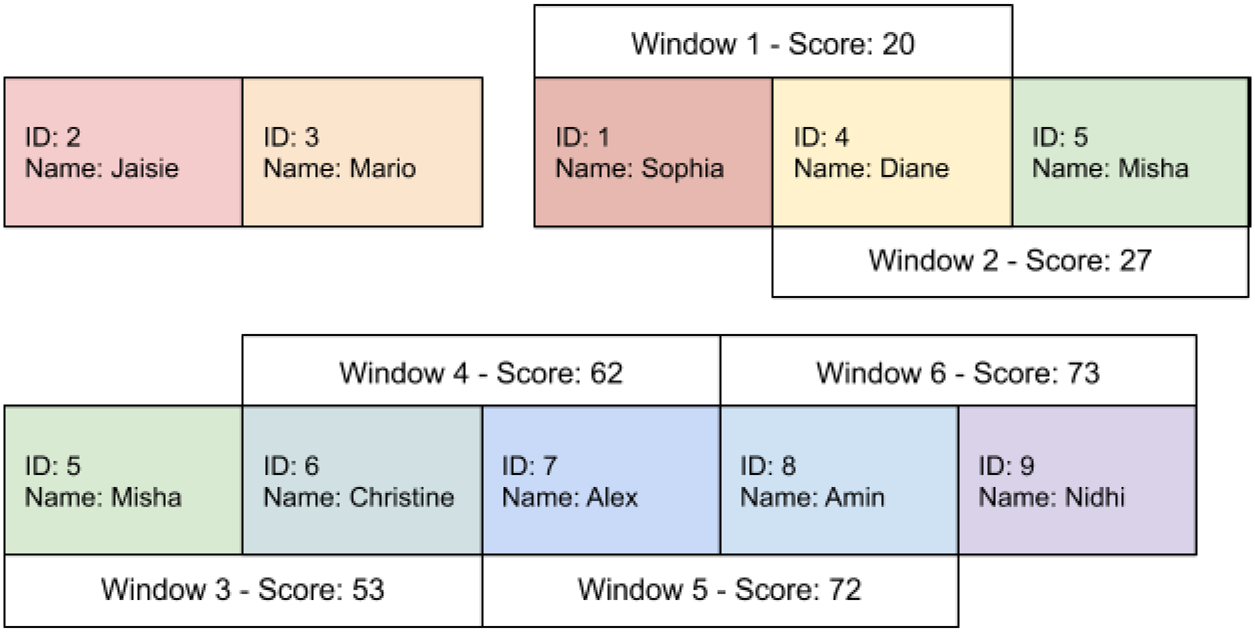
ID: 8
Name: Amin

ID: 9
Name: Nidhi

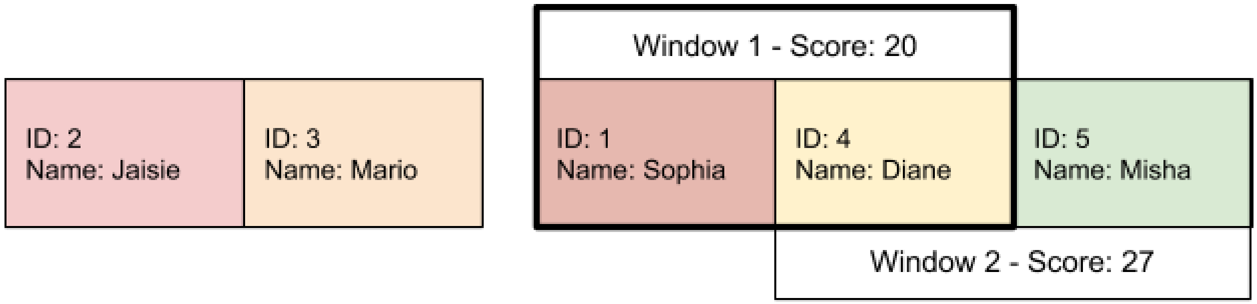
The remaining students are then placed in windows:

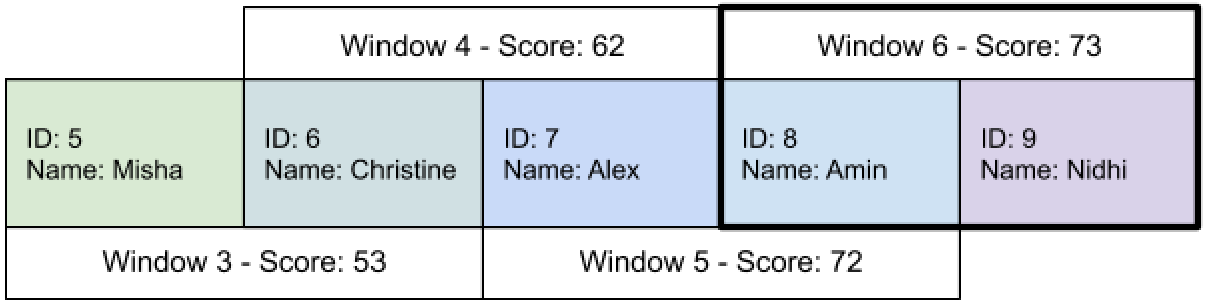


And we repeat the process, finding the first window that has a score greater than or equal to the window after it. Suppose the remaining windows have the following scores:

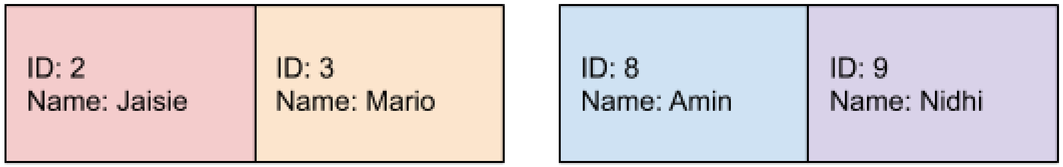


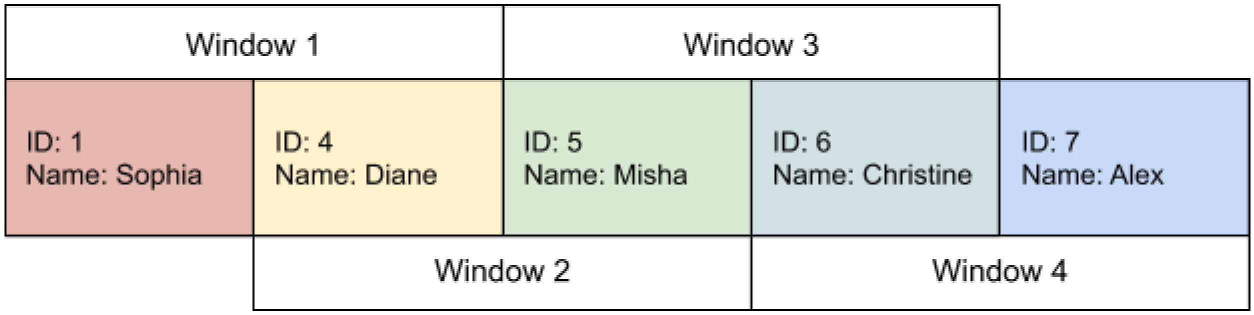
In this scenario, none of the windows have a score higher than or equal to the window after it, and thus we reach the last window. When we reach the last window, we then compare it to the first window:





Since the last window's score is higher than the first window's, that window is turned into a group, and we start the process again:





At the very end, if there are not enough students to make more than one window, those students are placed into a new group.