**Limiting involvement in project**

**Input:** student formal context, project formal context,

Lists of projects allotted to each student, limit\_val.

**Working:**

For each student, Calculate the significance for each project allocated to the student as per follow:

1. If student’s skills is not required in the project, assign significance[project] := -1 (least signif.)
2. If student’s skills requirement in the projects can be fulfil by other students included in the project, then assign

Significance[project] := total skills - no. of common skills between the student and other student in the project (no of common skill will decrease signif.)

1. if student possess unique skills in group that are required in project, then assign highest significance

significance[project] := total skills + no. of unique skills

remove the projects allotted to the student starting from least significant project to max. till project is less than or equal to limit\_val and project doesn’t requires student’s unique skills.

**Output:** Lists of projects allotted to each student.

Example output:

---------------------------------------------------------------------------

after limiting projects

---------------------------------------------------------------------------

project part ....

project1 ( 3 ) >{student3,student4,student8,}

project2 ( 3 ) >{student3,student7,student8,}

project3 ( 3 ) >{student2,student4,student10,}

project5 ( 2 ) >{student10,student5,}

project6 ( 3 ) >{student1,student6,student9,}

project7 ( 3 ) >{student2,student3,student8,}

project8 ( 2 ) >{student1,student7,}

project9 ( 2 ) >{student10,student5,}

project10 ( 5 ) >{student1,student2,student4,student5,student7,}

---------------------------------------------------------------------------

student part ....

student1 ( 3 ) >{project6,project8,project10,}

student2 ( 3 ) >{project3,project7,project10,}

student3 ( 3 ) >{project1,project2,project7,}

student4 ( 3 ) >{project1,project10,project3,}

student5 ( 3 ) >{project5,project9,project10,}

student6 ( 1 ) >{project6,}

student7 ( 3 ) >{project2,project10,project8,}

student8 ( 3 ) >{project1,project2,project7,}

student9 ( 1 ) >{project6,}

student10 ( 3 ) >{project3,project5,project9,}

**Plot dendograms for students and projects**

**Input:** formal context of students and projects.

**Working:**

do following for each formal context

1. calculate distance matrix for distance between each pair of students/projects that is number of different skills between students/projects.
2. Input the distance matrix to scipy.cluster.heirarchy.dendogram

**Output:** Dendogram

Example dendograms:

