If a schedule is not valid:

Compare the sets of required courses:

For all of them that are not in the courses already taken list or the generated planned schedule:

{

To insert a necessary course: Take the lowest intrinsic valued course that is not in the required category, and pop it out. Put in the required course where the low-valued, not needed course was.

Then check if this satisfies prereqs, if so, continue with loop, else, for each prerequisite, find next lowest intrinsic valued course that is not in the required category and pop it out, move the prerequisites into the earlier of these semesters, and the course that needed the prerequisite into the highest of the semesters.

}

##eofsubroutine

Method for entering courses to semester schedule:

{

find the two highest intrinsic value courses remaining to be taken. (frac function)

for each of these two courses:

determine which prereqs (the best set) are not satisfied (highest efficiency (being total of fractions divided by the number of them), then lowest number required)

determine the prereqs of those prereqs not satisfied in the same fashion

find the farthest down (longest “shortest path”) prerequisite, and add that to the schedule. (actually, take the two farthest down prerequisites for each of the two highest intrinsic valued courses) //this should allow a good averaging effect, and not too much focus

Then add the highest “no prerequisites left to take for it” course and add it.

//this simulates taking a “fun” course; this could also be replaced with adding an elective.

}

It would be good to have global lists of courses for the required areas: core, math, etc.

For concurrency, if a course has a concurrency needed as a prereq and doesn’t already have that prereq taken in a prior semester, then store it temporarily in a list, where if the concurrency course isn’t taken as one of the other courses, then empty this list after filling out the rest of the semester and take the next course (that or could maybe put it at priority one for the next semester, or something like that). When adding the other courses for the semester, check this list of “concurrency” courses

Therefore:

CheckPrereqs

{

Determine if all of the necessary prerequisites in a group of prerequisites have been taken. If not, check if all but concurrent flagged courses have been taken, and then add the course one is attempting to add to a list called “concurrency” that will be (a field… yes, it has to be a field)…………….continue later … evaluated when the last course needs to be added. What has been built of the schedule is checked and the concurrent course is added to the schedule if the needed course was in the up until that point constructed schedule.

//this is necessary because there needs to be temporary

}