Summary of Algorithm Going Forward

**Process Order:** Course Ordering 🡪 Semester Difficulty Optimization 🡪 Semester Hours Optimization

1. Courses are chosen by the user in selecting their tech core and either choosing electives or letting them be represented by placeholders with the average difficulty value of all available electives.
2. Construct a Directed Acyclic Graph from the selected courses with directed pre-requisites represented by directed arrows.
3. Produce all possible topological orderings from the graph to get every ordering of courses.
4. Iterate through each topological ordering, partitioning each set into N subsets with as even cumulative distribution in each subset as possible.
5. Now we will optimize for course hours as well. Divide the aggregate number of hours of all selected courses by the number of semesters; this will be our approximate goals number of hours per semester. Iterate through each of the, now partitioned, topological orderings and sum the absolute difference between the number of hours in each partition and our goal number of hours. Selected the topological ordering or the top 5 (or however many we deem necessary) with the least cumulative partition-hour-difference.

**Conclusion:** Multiple parts of this algorithm are actually NP-complete problems such as the “partitioning problem” or attempting to partition a set into N partition with minimal different between them. Although, producing every possible topological ordering is a brute force method and traditionally inefficient, it will work for the task at hand because of the bound of the number of permutations resulting from ECEs fairly strict course structuring and the limited number of classes one can take in a college career.