1. Possible Project Topic #1 (OUTLINE)
   1. Necessary resources:
      1. Resource 1
      2. Resource 2
      3. ...
   2. Intended demographic (who will be using/seeing/benefitting from this project)
      1. Value to user
      2. Plan to make project available to audience (Presentation, flyer, Internet?)
   3. Pros and cons for programmer
      1. PROS
         1. Pro 1
         2. Pro 2
         3. ...
      2. CONS
         1. Con 1
         2. Con 2
         3. ...
   4. Demonstrated skills/knowledge:
      1. Skill 1
      2. Skill 2
      3. …
2. Essay on Data Structures and Algorithms
   1. Necessary resources:
      1. Google Docs
      2. Peer editor
      3. Online Sources
   2. Teachers/mentors, few students
      1. Explains subject and demonstrates understanding of content
      2. Printed out, on Google Docs, put on website?
   3. Pros and cons for student
      1. PROS
         1. Fairly straightforward
         2. Requires little creative deviation
         3. Reliable
      2. CONS
         1. Written in English, not a programming language (so it is less impressive)
         2. Time consuming
         3. DUll for the reader
   4. Demonstrated skills/knowledge:
      1. Understanding of data structures
      2. Knowledge of algorithms
      3. Comparison and contrast/analysis of problem solving techniques
3. Implementation that combines all ADTs learned and performs different operation methods (e.g. insert, delete) and algorithms (e.g. traversals, sort) when called
   1. Necessary resources:
      1. Python text editor (likely Sublime)
      2. List of data structures, example implementations (online bookmarked and in notes already)
      3. ...
   2. Intended demographic (who will be using/seeing/benefitting from this project)
      1. User can interact with program to learn or use data structures
      2. Put downloaded version on website or try to put ON website if possible (https://docs.python.org/2/howto/webservers.html), GitHub (maybe Gist?) and give to teachers Keylon, Jacobson, etc
   3. Pros and cons for programmer
      1. PROS
         1. Allows me to demonstrate knowledge in context
         2. Can return to as a way of studying
         3. Can be used as a fun or valuable tool for understanding or organizing data
      2. CONS
         1. MANY implementations to choose from
         2. Account for debugging time/stress
         3. Lots of data structures, algorithms to cover (can’t do all)
   4. Demonstrated skills/knowledge:
      1. Ability to code implementations of different data structures
      2. Understanding of different operations to be called according to the DS
      3. Ability to develop a program with a user friendly interface (though this may end up being totally text based/i.e. An independent python file)
4. A web page on which the user can click buttons to filter different data structures and algorithms based on their characteristics (e.g. “abstract,” “primitive,” “data structures,” “linear data type,” “algorithms,” “search algorithms,” “computational complexity O(log n),” “minimum spanning tree algorithms,” “plain text implementation code” (which would be based on my previous projects) etc)
   1. Necessary resources:
      1. Online sources/notes
      2. Mentor to double check accuracy
      3. Website domain to house project (already have)
   2. Anyone with access to the Internet and wants to learn about data!
      1. Teaches the distinctions b/w DT & A in a fun and intelligible way
      2. Internet (hosted on my website)
   3. Pros and cons for programmer
      1. PROS
         1. Fun, and a change of pace from programming to markup languages
         2. Useful study tool
         3. Immediately accessible to me, and a neat addition to my website
      2. CONS
         1. Very data intensive
         2. Less about programming new implementations and more about displaying
         3. May new to find implementations which I have not programmed myself (can cite source at button, “SOURCE: Crista Falk” etc)
   4. Demonstrated skills/knowledge:
      1. Extensive knowledge of DS & A
      2. Understanding of the important distinctions and similarities of “ ”
      3. Ability to program in python (as implementations convey) and JS, HTML/CSS (as the website itself shows)

OTHER PROJECT IDEAS:

-- Make a text-based game

-- Make an Enigma Machine

-- Build a "smart" calculator that allows for undo and redo actions.

-- Build a 1-player checkers game with a basic AI opponent.

-- Make a Haiku generator

Basically, the Synthesis Project should be an opportunity to demonstrate where certain data structures are useful; it's a "ride the bike" rather than the "read the manual" approach.