TODO:

- Understand the Pythagorean Triplet algorithm - derivation from primitive Pythagorean Triple "formula"
- Optimization of Collatz Conjecture Sequence (maybe -> using existing nodes)
- Extensions to fun questions:
  - Collatz Conjecture -> create a graph relating odd numbers to their next odd node -> from (1, 1000)
    - Frequency graph
  - Program (visualization) factors being generated from a number using the prime factorization algorithm: find prime numbers upto n/2 (p1, p2..., pn) -> divide n/p(n) if (n%p == 0)

General:

- Better understand every "optimization" found within the Euler page
- Go through current iPython Notebooks to find optimizations for each algorithm (-> significant optimizations)
  - Factors: currently iterating/creating prime factorization for each triangle number (p1^e1 * p2^e2 * p3^e3... * pn^en) and finding amount of factors using ((e1+1) * (e2+1) * (e3+1)...(en+1))
  - Maximum Path Sum: find a more effective method of generating orders -> 2^n (currently iterating through numbers upto 2^n+1 and going through (if) statements... (inefficient)) -> find more effective way of getting solution -> comparing values?