Implementation
Details

- Library functions have been used whenever possible.
- A new custom type has been defined in order to deal with the custom weight function used in Algorithm WD. This new type subclasses `tuple` and implements addition and comparison as described in the paper.
- Time complexity has been assessed with the library big_o, which just measures the running time for different values of n and then fits a set of different time complexity classes, in order to find the best one.
- Space complexity has been assessed analytically, due the absence of a Python library able to do an empirical analysis.

Implementation

Time complexity assessment

- Algorithm CP [O(E)]
  Linear: time = 0.00033 + 1.2E-05*n
- Algorithm WD [O(V^3)]
  Cubic: time = 0.039 + 3.8E-07*n^3
- Algorithm OPT1 [O(V^3 lg V)]
  Cubic: time = 0.066 + 4.7E-07*n^3
- Algorithm FEAS [O(VE)]
  NODES: Quadratic: time = 0.014 + 2.8E-05*n^2
  EDGES: Quadratic: time = 0.0051 + 1.5E-05*n^2
- Algorithm OPT2 [O(VE lg V)]
  Polynomial: time = -9.3 * x^2.2