Simulated Annealing Pseudo Code:

1. Generating the Graph G with edges values ranging from 4000 to 10000
2. Initializing the staring path as p and temperate as t
3. Initializing a for loop for t=12000 to 0:
    a. If t==0:
        i. Break
    b. Else:
        i. the temperature is scheduled using the following function
            1. $t= \alpha t/\ln(1 + k)$ where $\alpha=1$
        ii. picking Path P of G consisting source and sink
        iii. Changing the edges of path P by adding or subtraction 3000 which gives the Graph G'
        iv. If check_constraint== true:
            1. Delta E= value(G)-value(G')
            2. If Delta E > 0:
                a. G=G'
            3. Else:
                a. Var,Rand_num, acceptance_region = convert_probability()
                b. If the rand_num lies in the acceptance region:
                    G=G' with probability var

    Return solution G

4. The check_constraint takes the Graph G and path P and check for the following:
    a. Law of Conservation, the sum of the out flows from the node cannot exceed the sum of the inflows into the same node
    b. Edge constraint, the actual flow a node should be less than or equal to the capacity of the node.
5. The Value() takes the Graph G and returns the sum of inflows of the sink node n
6. The convert_probaiblity function takes the delta E and t and is implements as
    a. $Var=e^{\text{delta } E/t}$
    b. Using math.frexp(var) to get mantissa and exponent
    c. Using the random function to select number rand_num from the interval 1 to 10*exponent and the acceptable region stretching from 1 to mantissa.
    d. Returns var, rand_ num, acceptable region,