NAME

SimulatedA

DESCRIPTION:

This module contains the algorithm for Simulated Annealing + Two Opt

FUNCTIONS

randomIndexes(rand)

:param rand: range for random.randrange()

:type rand: integer

:return: index i and index j

:rtype: integer

sA(startRoute)

:param startRoute: route to improve

:type startRoute: Route

:return: returns an improved version of the startRoute and the time it took to find this improved version

:rtype: Route, float

:warning: not enough connections between each city/point can lead to the algorithm becoming less effective

swap(r, i, j)

:param r: route

:param i: index

:param j: index

:type r: Route

:type i: integer

:type j: integer

:return: void, function swaps the values on index i and index j with eachother

swapCheck(newR, i, j)

:param newR: new route

:param i: index i

:param j: index j

:type newR: Route

:type i: integer

:type j: integer

:return: void. Function will keep swapping values until a valid route has been made

---

Pseudo Code Simulated Annealing

```
startRoute <- random route
currentBest <- startRoute
bestRoute <- startRoute
Temp <- 1000

While (Temp > 1) do
        newRoute <- copy(currentBest)
        swap two points in newRoute
        difference <- (distance currentBest – distance newRoute)
        acceptance < 1/(10+(difference/Temp)^2)

        if difference > 0
                currentBest <- newRoute
        else-if acceptance > randomNum(0, 1)

        if distance currentBest < distance bestRoute
                bestRoute <- currentBest

        Temp <- Temp – 1
```