CS 404 – Artificial Intelligence
HW 3 – Local Search

Goal: To learn more about local search which has very low memory usage and can be quite successful for many problems; and to gain further experience with programing and reporting research results.

Task: Solve the N-queen problem with different local search algorithms. N-queen problem is finding the placement of N queens on an NxN board such that noone attacks one another. The given Python code implements basic hill climbing and random restarts. You will be asked to a) run simulation experiments with the given code and b) expand the code with stochastic hill climbing.

a**) 50 pts**- Given the code in link , run **100** simulations/experiments with different initial solutions and give the number of **successes, number of iterations, and the time it takes** to find the solution on **average** in each case, for **N=10 and N=20,** for **basic** hill climbing and **random restart** with increasing number of restarts (k=10, 100, 1000)

	N=10			**N=20**		
	Percentage of success in 100 runs	**Solutions found in how many restarts on average**	**Elapsed time to complete experiments**	**Percentage of success in 100 runs**	**Solutions found in how many restarts on average**	**Elapsed time to complete experiments**
Basic Hill Climbing	8%	-	0.0089	2%	-	0.2524
Random Restart with k=10	41%	4.76	0.0698	11%	4.82	2.3188
Random Restart with k=100	100%	15.47	0.1376	87%	31.77	9.7961
Random Restart with k=1000	100%	14.99	0.1316	100%	49.53	11.8917
Stochastic hill climbing *(to fill for part b)*	8%	-	0.0124	2%	-	0.4136
Simulated Annealing *(to fill for part c) – if you will do the bonus)*	6%	-	0.1886	3%	-	2.9761

b) **50 pts**- Add a new function randomNeighbor(....) to implement **stochastic hill climbing**. If no better neighbour, should return current one. Leave other code the same.

Fill the results of 100 experiments to the corresponding row of the table with stochastic hill climbing.

c) Bonus-**15 pts**: Implement **simulated annealing** and fill the results of 100 experiments to the corresponding row of the table. Specify the best parameters you found (what is the schedule and initial temperature) here or in the table.

The best parameters I found for simulated annealing are as follows:

- Initial temperature = 100000
- Annealing rate (schedule) = 0.9

How I found these values are explained in my colab file which its link can be found below.

https://colab.research.google.com/drive/112NIHHGowovSbEeHEJ4mMhNQod6OxsO5?usp=sharing