Christian Lei
Assignment 3
11/9

1. What happens if we don't allow for sideways moves? (relates to part 3)
   The purpose of this lab is to minimize the objective function, which means that we are finding a way to put the most moves from start to goal in the board. To do this, we constantly switch single digits of the board and then accept them if they produce an equal objective result or better. Taking the equal objective result means allowing sideways moves. This helps prevent the algorithm from getting stuck in a plateau or shoulder when there are no more downhill moves. If we didn't allow for sideways moves, then we would get stuck in one of these situations, since each side is of the same value, and then we would miss out on finding the global minimum that we are looking for.

2. What is the impact of the allowed number of consecutive sideways moves on the hill-descent algorithm? (relates to part 3)
   What I see as consecutive sideways moves is the number of iterations. If we were to not limit the number of sideways moves, the algorithm could get stuck on a flat maximum since we're technically at a maximum, but each side is the same value and we could end up going back and forth infinitely. This iteration count prevents that. Once we reach the iteration count, we return the value and that is our answer. The more moves there are, the more chances we have to escape that flat area. It might take the algorithm longer, but it allows for a higher chance of success. Thus, you need to take these factors into account when allowing the sideways moves.

3. Keeping the number of allowed sideways moves constant (say 10000), what is the effect of the number of restarts? (relates to part 4)
   The effect of the number of restarts is to maintain the objective value that we have collected thus far, but to start again in another area of the landscape. In effect, it suggests that we may be in a local minimum and this is one way to escape. One common problem with hill-descent search is that it gets stuck in local minimums, since the steps from that minimum are all higher and the algorithm won't naturally go to those steps. By randomly restarting, we get a new perspective and hopefully have escaped that local minimum and we can find the global minimum. The more restarts that there are, the more chances we have to find the global minimum by starting in a different area of the landscape.

4. Is the probability of taking an uphill step the same as the probability of escaping a local minimum? Why or why not? (relates to part 5)
   No, I do not think that it is. The one uphill step might not get the algorithm out of the local minimum. You might need to take multiple steps to get out of it, and thus, you would have to multiply the probabilities of the two steps together and that probability would be lower than the probability of the one step.

5. For a fixed maze with a fixed number of iterations (say 10000), what is the best annealing schedule? Run multiple simulations with different annealing functions and compare and contrast them. (relates to part 6)

   Equation for probability: $e^{-\Delta E/T}$
   Probability of taking a "bad move" increases when the exponent is small. There is also a random chance of anything happening since it depends on the board that is created.

Annealing Schedule: Linear
Avg = -9.67

Annealing Schedule: Exponential
Avg = -8

Annealing Schedule: sqrt
Avg = -10

For this simulation, I created three types of annealing functions. One linear, one logarithmic and one that is the square root. It seems that the square root produced the best results while the exponential produced the worst and linear was in the middle. I would say that it is hard to compare and contrast them since they are not consistent data points and everything is random. But these averages are from multiple runs.

I think that the square root produced more consistent results. It seems to me that the slower the temperature decreases, the more consistent the results are. But it also seems to me that any of the schedules can produce a good result, it really depends on the luck of the randomness.