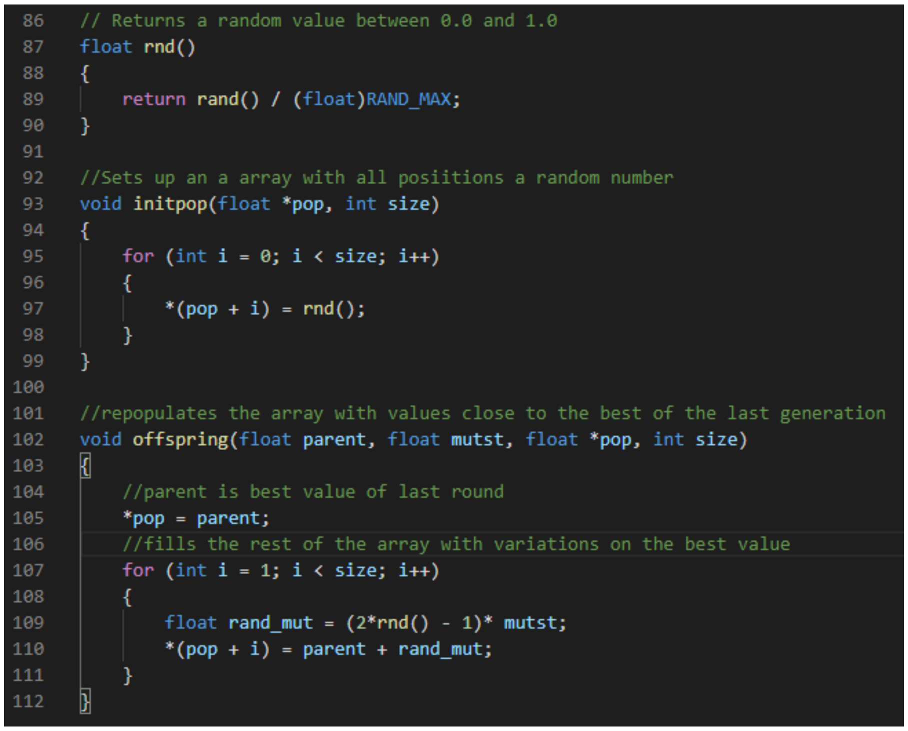
What part(s) of Darwin's algorithm (Reproduction, Variation, Selection) is happening in the for-loop inside main()?  I Write down the condition needed in the if-statement inside main().

Selection.

Code for initpop and offspring, filled in the skeleton code.



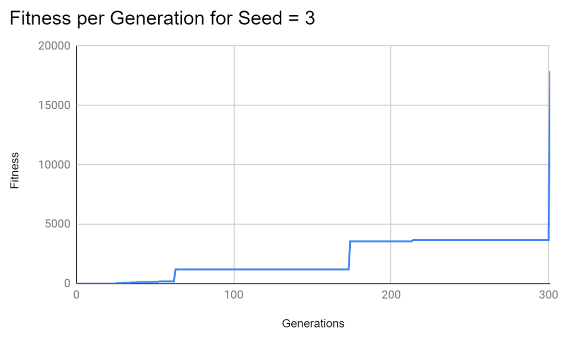
```c
// Returns a random value between 0.0 and 1.0
float rnd()
{
    return rand() / (float)RAND_MAX;
}

//Sets up an a array with all posiitions a random number
void initpop(float *pop, int size)
{
    for (int i = 0; i < size; i++)
    {
        *(pop + i) = rnd();
    }
}

//repopulates the array with values close to the best of the last generation
void offspring(float parent, float mutst, float *pop, int size)
{
    //parent is best value of last round
    *pop = parent;
    //fills the rest of the array with variations on the best value
    for (int i = 1; i < size; i++)
    {
        float rand_mut = (2*rnd() - 1)* mutst;
        *(pop + i) = parent + rand_mut;
    }
}
```

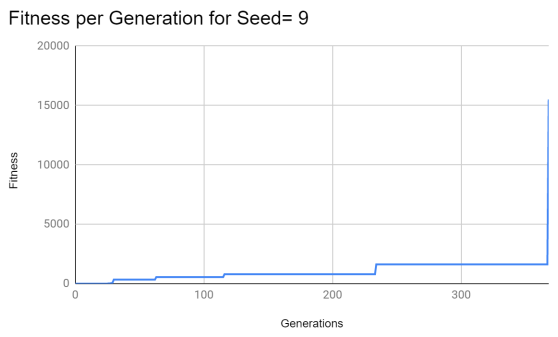
**21/10/20**

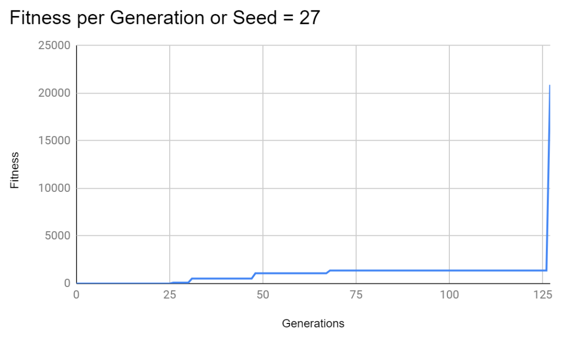
I have modified my equation to be $y = x^3 - 42$, I will run this equation with different seeds to find the value of the cube root of 42.

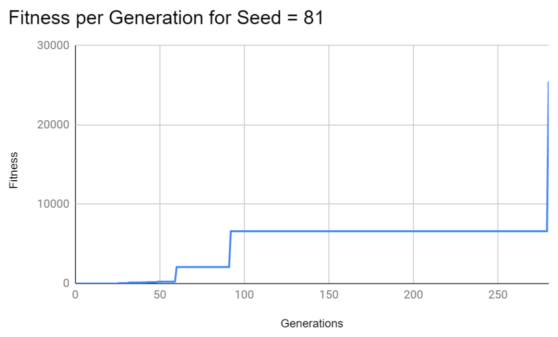
| Run | Seed | Generations | Solutions |
|---|---|---|---|
| 1 | 3 | 301 | 3.476025 |
| 2 | 9 | 368 | 3.476025 |
| 3 | 27 | 127 | 3.476028 |
| 4 | 81 | 280 | 3.476028 |
| 5 | 243 | 546 | 3.476027 |

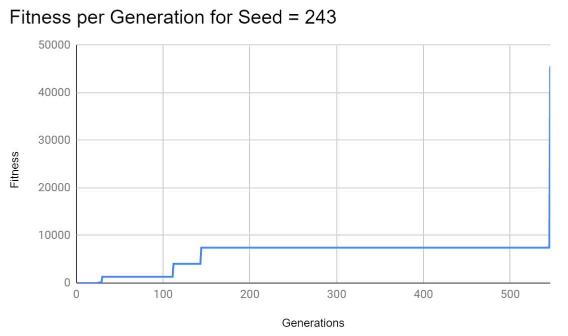
All the seeds will find the root within acceptable tolerance from the true value. I found that the value is between 3.476025 and 3.476028, with the true value being 3.4760266…. Meaning it is accurate to 6 significant figures.





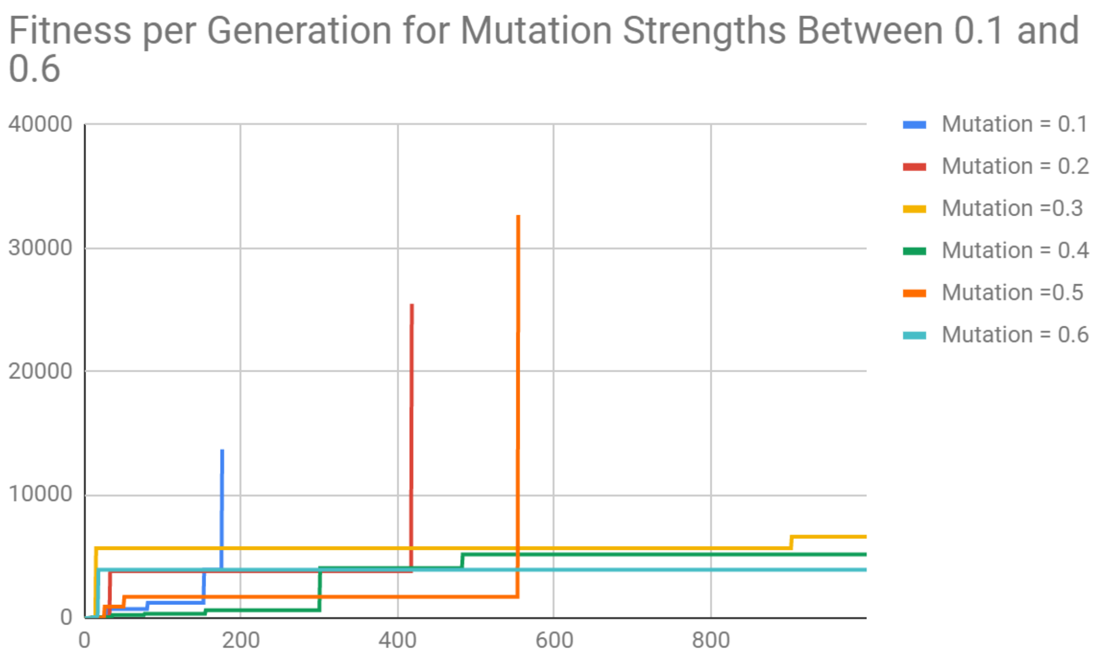




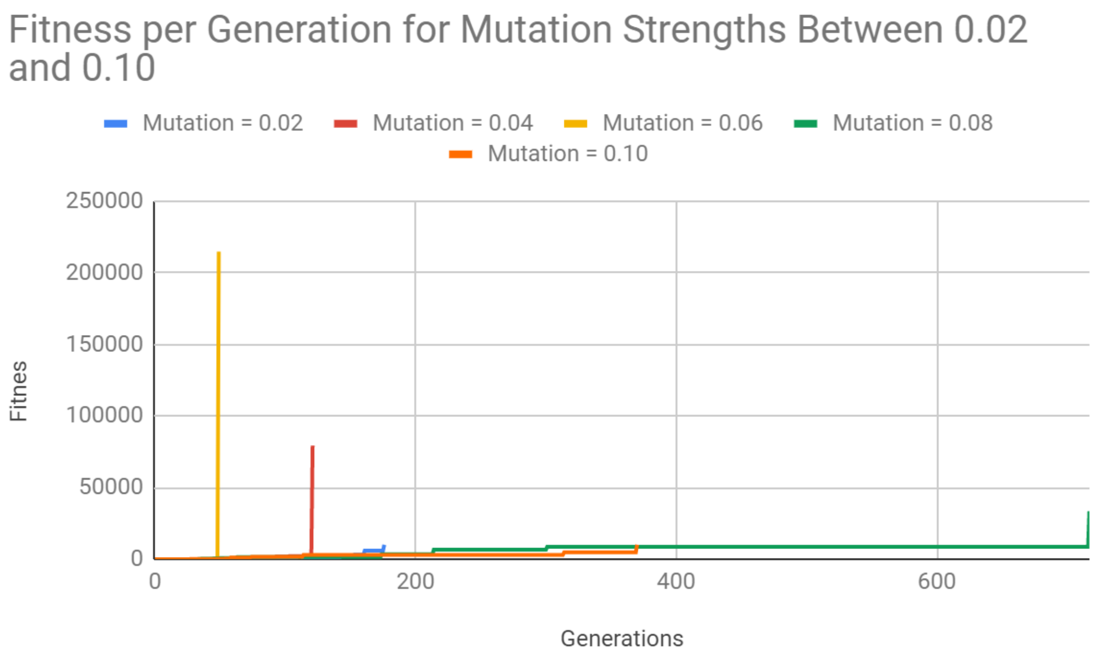


I found that there is a slow incremental increase for long periods. Then for small periods of a couple generations where massive improvements are made. I would guess this is because the programme gets stuck on a sub-optimum path for a long time and is waiting for a mutation to let it break through.

When investigating the effects mutation strength has on how long it takes the evolution program to find the (good enough) root I expect that too small a value then it will take long because it can only make incremental steps, but too large a value and it will fluctuate too much and will only be able to find the value by luck.

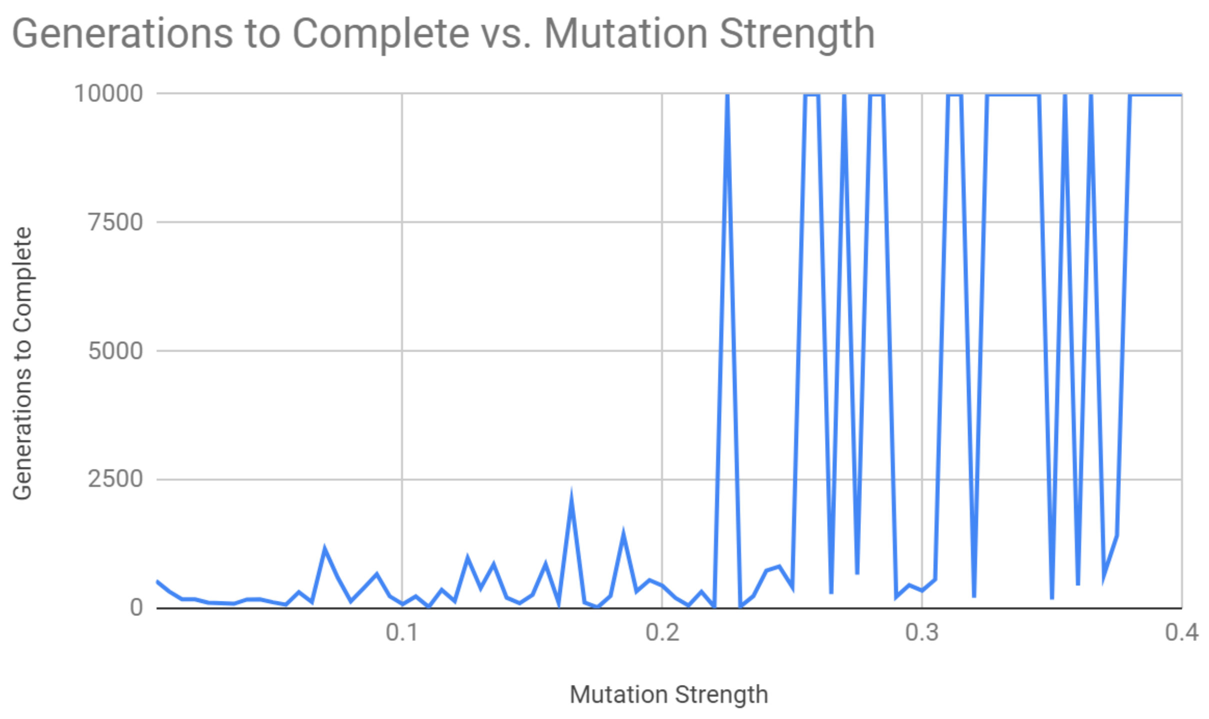


I found that for larger values of mutation strength it takes longer to find the value. I expect mutation strength = 0.5 is an outlier as it had a consistently low fitness and only found the value by chance.



I found that with the smaller values that the mutation strengths between 0.02 and 0.06 found their solutions much faster than the higher values. From this I would suggest that the mutation strength that takes the least generations would be around 0.05.

By editing my code I was able to make it print out the number of generations it took for each value of mutation strength.



With this program I tested a large sample size of mutation strengths and I found that a value below 0.2 would reliably find a solution. However above 0.2 the mutation Strength is high enough that it is unable to consistently find the root.

*“We can take a different perspective on our population. Let’s assume the individuals in our population (the array of floats) are fashion-conscious and hungry for attention. Limit the possible characteristics (float value) of the individuals to be in the interval 0.0–1.0. If three of them meet, then whoever stands out most in the group wins. To do this we calculate the difference between any pair in the group of three, and sum for any individual in the group the differences. The individual with the largest sum wins the round. As a reward a (small) number of followers are created by copying the winning individual with small random variations (staying within the 0.0–1.0 bounds). These followers are placed back into the array at a random location (i.e. they replace random other individuals). What do you expect to happen if we repeat this process indefinitely? “*

In this scenario I anticipate that the array will homogenise onto one or two of the fittest values. There will be small variation on this (or these) values that could lead to small increases in fitness but I anticipate as time goes on these will become rare. If one of these groups get enough incremental increases then in a few generations it will wipe out the other group.