EVOLUTION: NATURE'S LEARNING ALGORITHM

PRESENTATION BY: MAGDALYN ELKIN

“

All knowledge – past, present, and future – can be derived from data by a single, universal learning algorithm.

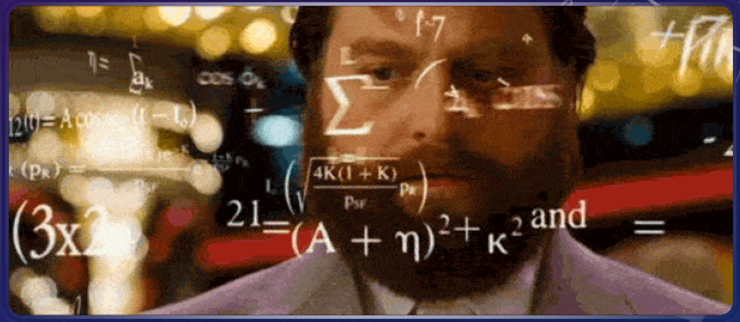
”

Pedro Domingos

The search for the master algorithm takes many forms from different tribes of machine learning. This presentation will focus on evolutionaries and genetic algorithms.

GENETIC ALGORITHMS

- Artificial neural networks (ANN) are a common machine learning algorithm, that can be applied in many different fields
- ANN are a representation of neurons in the brain
  - They learn in similar ways that the brain does, by strengthening and weakening different input connections to produce a desired output
- Genetic algorithms are a representation of evolution
  - Evolution is natures most widely known optimization algorithm
  - Genetic algorithms approach mathematical optimization
    - How can we maximize or minimize a certain value?



https://media.giphy.com/media/3o6Yg4GUVgIUg3bf7W/giphy.gif

THE ORIGIN OF GENETIC ALGORITHMS

- In 1959 John Holland earned the world’s first PhD in computer science
- During his studies he was inspired by the work of Ronald Fisher’s *The Genetical Theory of Natural Selection*
- But he felt that the theory left out the essence of evolution, which includes key components of natural selection
- He created the first genetic algorithms which included fitness functions

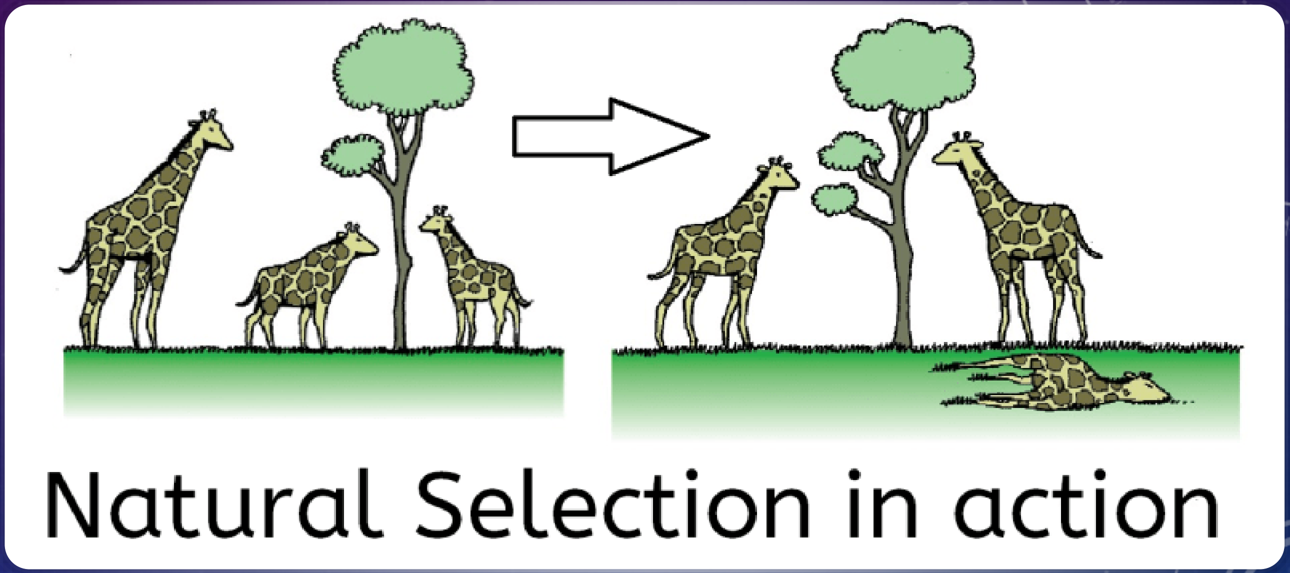
“The fact that life evolved out of nearly nothing, some 10 billion years after the universe evolved out of literally nothing, is a fact so staggering that I would be mad to attempt words to do it justice.”

Richard Dawkins

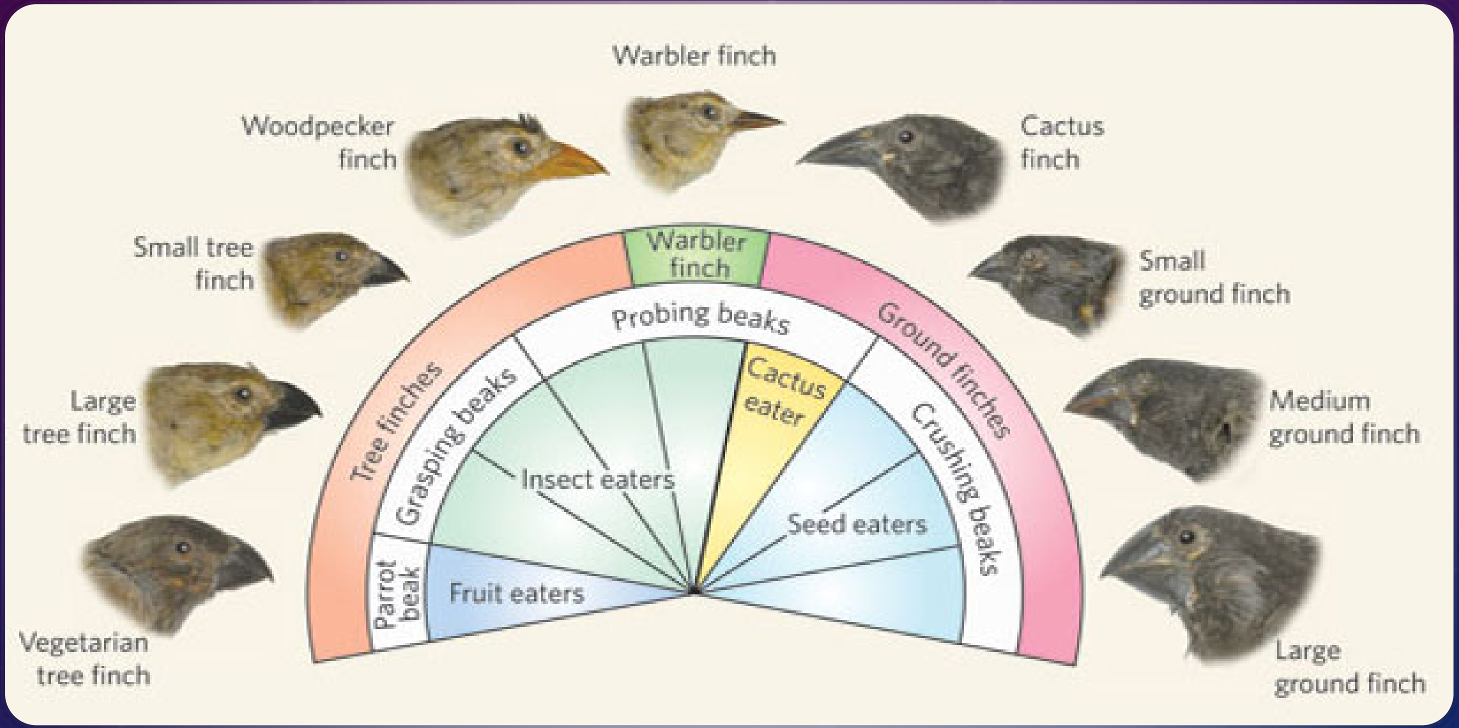
To describe genetic algorithms, we first must learn key concepts of evolution. These are the foundations for genetic algorithms

DARWIN'S ALGORITHM

- In 1859 Charles Darwin wrote On the Origin of Species
- This introduced the theory of natural selection
  - Organisms better adapted to their environment tend to survive and produce more offspring
  - After generations of selecting and mating, the organisms we see today are products of those that best served their purposes



https://www.createwebquest.com/natural-selection



https://www.thinglink.com/scene/495598280160313345

DARWIN

- Darwin's theory of natural selection came about from observing different finch species in the Galapagos islands
- He noticed that different species of finches lived in different areas of a tree.
- These species had different shaped beaks and different diets
- Finches that ate insects had grasping beaks to quickly snag an insect
- Finches that ate seeds had crushing beaks to break apart seeds

COMPONENTS OF NATURAL SELECTION

Heredity

Children receive the properties of their parents

Traits are passed from parents to the children to their grandchildren

Variation

There is a variety of traits in the population

Without variety then children will always be identical to their parents and no new combinations will occur and nothing will evolve

Selection

Survival of the fittest

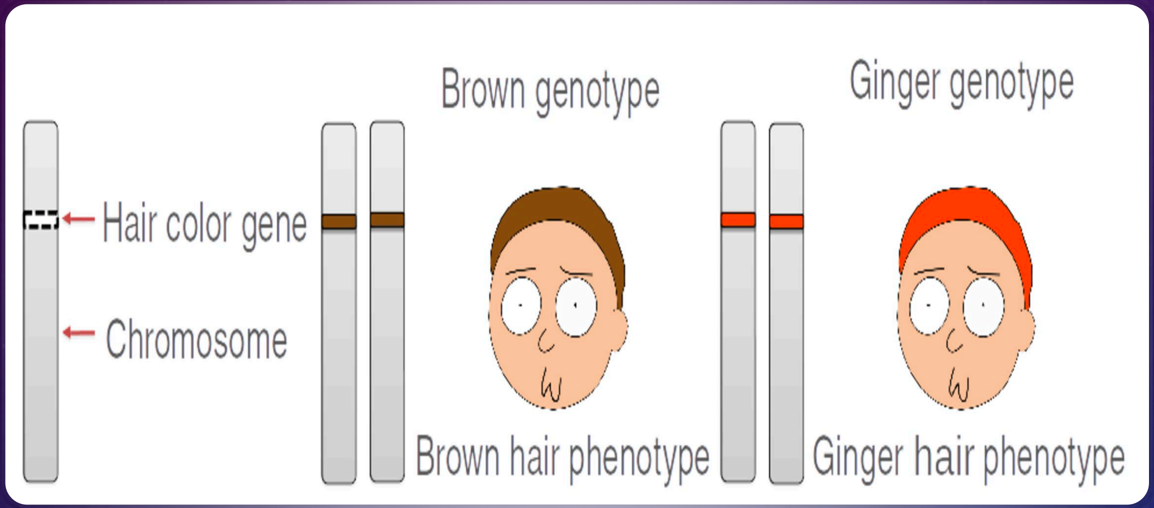
Some traits are better adapted for creature's environment and produce a greater likelihood of surviving and reproducing

HEREDITY

- Genes in your body are made of molecules of DNA that provide the instructions for your physical traits.
- All the genes in your body have been passed down to you from your parents and from their parents and so on
- Chromosomes are made of up genes that carry the genetic information
- We each have two chromosomes, one from our mothers and one from our fathers
- Parents each pass one a chromosome to their child, which carries all the genetic information required to generate physical traits that we inherit from our parents

https://www.cartoonstock.com/directory/h/heredity.asp

GENOTYPES AND PHENOTYPES



https://www.crowdai.org/challenges/opensnp-height-prediction

- Chromosomes carry the genetic code of our bodies
- The genetic code is made up of 4 DNA bases
  - A,T,C and G
- These bases have the “blueprints” for our body
- The genetic code is read and specifies which proteins to create
- These proteins generate our physical traits
- A Genotype is the set of genes an organism carries
- A Phenotype is the observed physical characteristics of the genotype

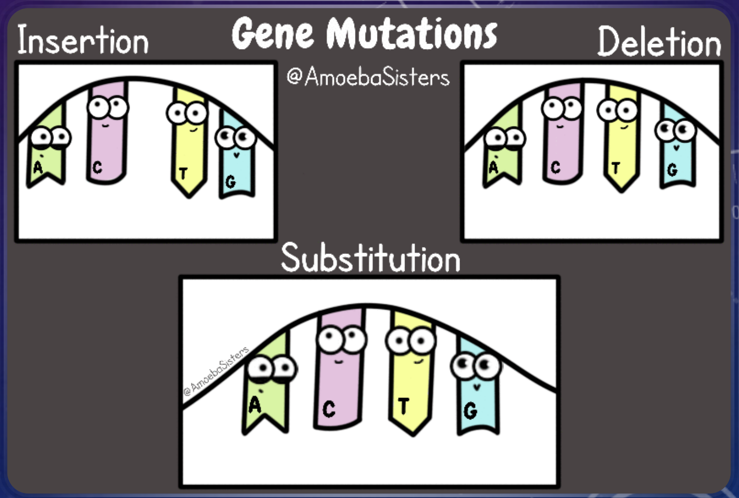
VARIATION

- While our genes are inherited from our parents, we don't carry the exact same traits as our parents
- Part of this is due to chromosomal crossover
- When a female and male reproduce, they each provide one chromosome to their child.
- These chromosomes undergo crossover where the resulting child's chromosome is a combination of his mother's and father's chromosomes
- This introduces slight variation in the offspring
- When the offspring in turn reproduce, the genetic material they pass on then will be bits and pieces of their parent's genetic material

http://fuzzysynapse.com/illustrations/

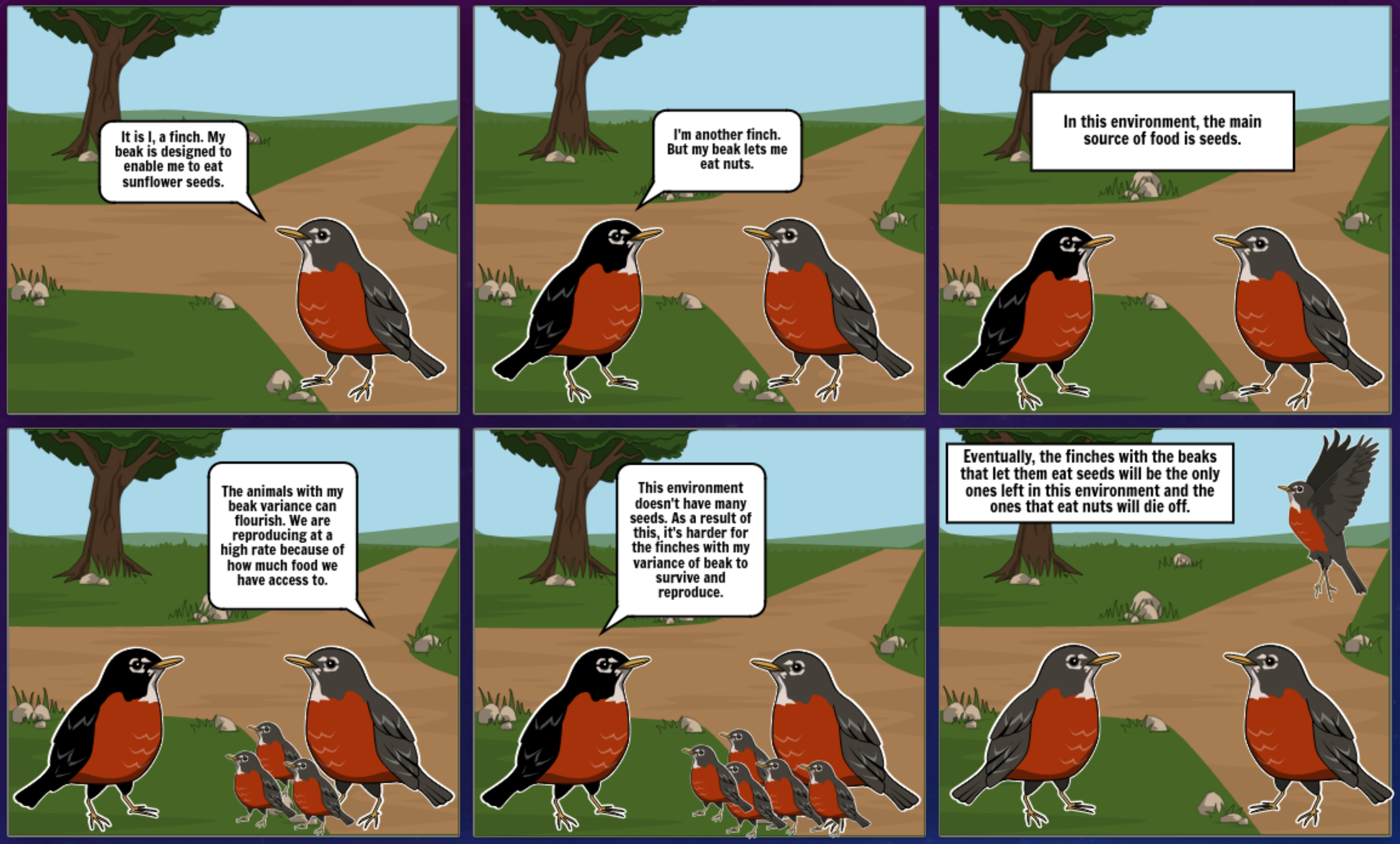
VARIATION

- Variation can also be introduced by genetic mutations
- The main genetic mutations:
  - Insertion
  - Deletion
  - Substitution
- These mutations change the DNA code of your body
- Genetic mutations can happen in many ways
- In some cases, genetic mutations are caused by external forces
  - Such as staying out too long in the sun! Wear sunscreen!
- In other cases, genes can randomly mutate which produce a new genetic combination
  - New genetic combinations can be damaging to the cells
    - Such as certain genetic diseases, like achondroplasia (dwarfism)
  - They can also introduce new traits to the population
    - Thru random mutations new physical traits can emerge that can potentially lead to the organism surviving better in its environment



https://i.pinimg.com/originals/42/3d/08/423d08d83e86ee136ea37cc4262826c2.gif

NATURAL SELECTION



Create your own at Storyboard That

https://www.storyboardthat.com/storyboards/6293d1cb/natural-selection

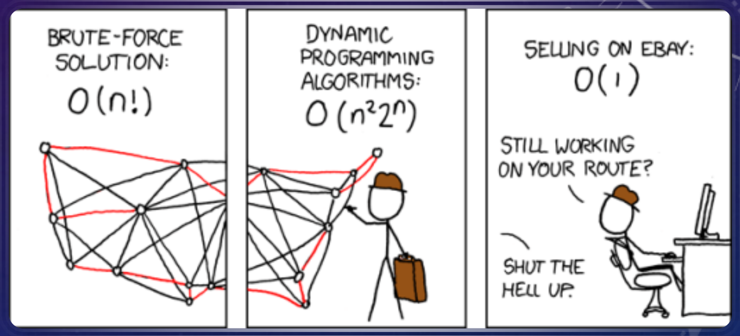
- Survival of the fittest
- Organisms that have traits better suited for their environment are more likely to survive and reproduce and pass their traits down to their offspring
- Meanwhile with variation mechanisms, new traits are being explored and potentially becoming the new “fittest” traits

GENETIC ALGORITHMS

- As we explore genetic algorithms, we'll see that all components for evolution are present in genetic algorithms

TRADITIONAL GENETIC ALGORITHMS

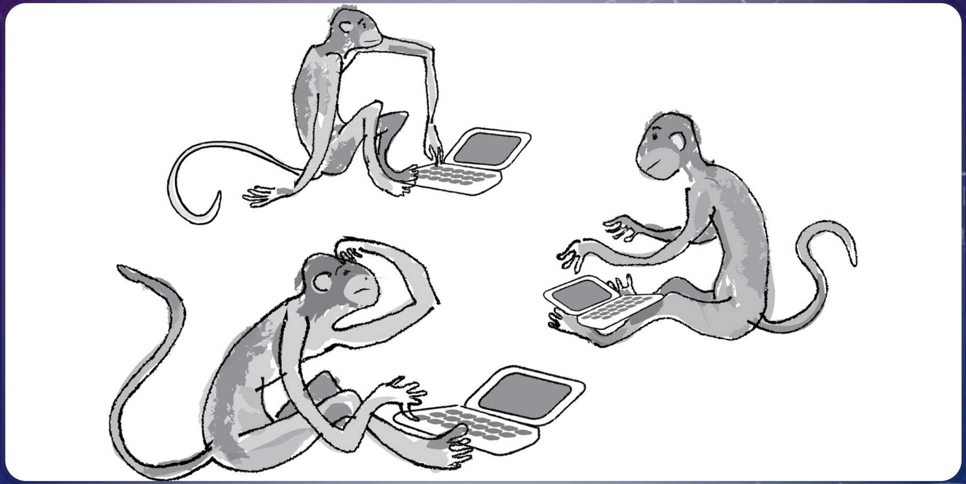
- The traditional genetic algorithm was created to solve problems where brute force algorithms would take too long
- Some problems have solution spaces so vast that a brute force algorithm would take forever to solve.
- Brute force algorithms run through every possible solution and tests it to see if it's correct
- If an algorithm had to guess a number between 1 and 1 billion and it used brute force, then it would take up to 1 billion time steps to guess it correctly
    - In reality, it would take less than 1 billion unless the algorithm was very unlucky
    - But in running time speed you always consider the very unlucky scenario
- If you could put in a method to check each answer how close it was to the correct answer, then you can arrive to correct answer in less time
    - In this way you could measure the "fitness" of your algorithms output and use that to guide to the correct output



https://techutils.in/blog/2016/02/26/travelling-salesman-problem/

INFINITE MONKEY THEOREM

- This theorem illustrates the traditional genetic algorithm
- Monkeys are tasked with typing out the complete works of Shakespeare
- A monkey furiously typing away will eventually reproduce all of Shakespeare's work
- Theoretically, there are only a finite combinations of words that can be created by typing on a keyboard, so even randomly hitting keys, the monkey is bound to eventually reach Shakespearean works
- Realistically, the probability of a monkey typing out Hamlet is so low that the monkey would still be working when the universe ends
    - Theorized to be somewhere between 2 billion and 22 billion years from now!



The Nature of Code figure 9.1

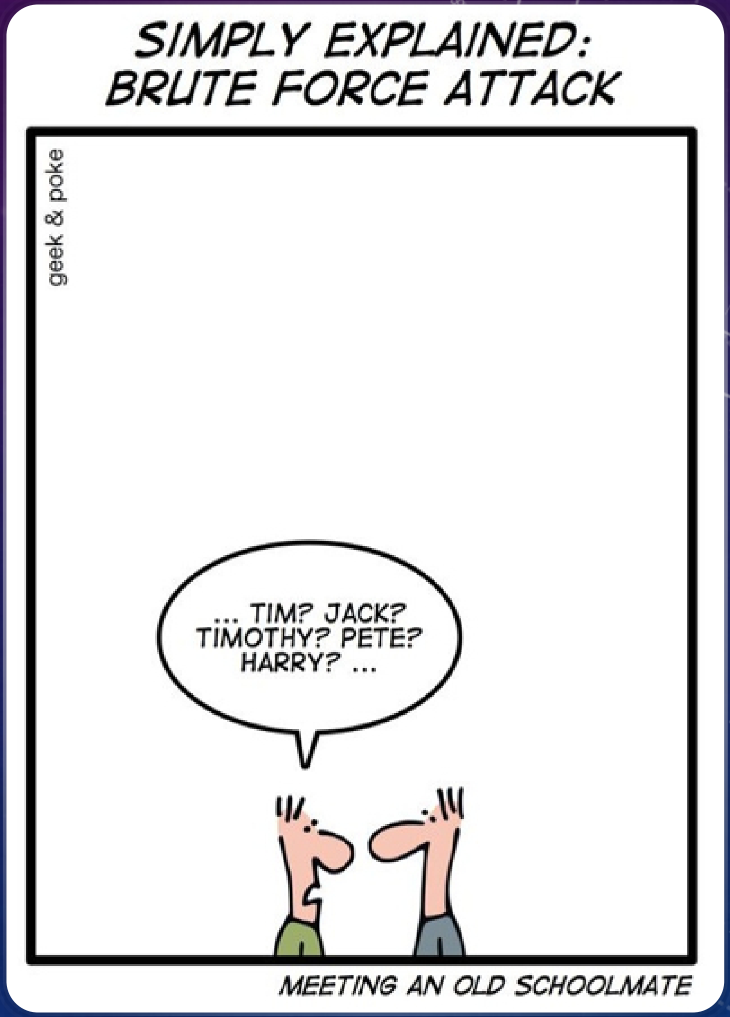
INFINITE MONKEY THEOREM

- Let's consider one monkey in particular, George
- George is tasked with typing out Hamlet.
- He is typing with a keyboard with 26 letters and a space bar only
- The probability he will get 1 character correct is 1/27
- He is trying to type out "To be or not to be"
- The phrase is 18 characters long
- Each character has 1/27 chance of George randomly stumbling across it
- Which means George has a $(1/27)^{18}$ chance of getting it correct
  - This is roughly 1 in 58 septillion chances
  - For context, the probability of winning the lottery is 1 in 14 million chances

https://www.npr.org/sections/13.7/2013/12/10/249726951/the-infinite-monkey-theorem-comes-to-life

WHERE BRUTE FORCE ALGORITHMS GO WRONG

- Brute force algorithms can be used in many situations
  - They can typically work if you have a small solution set to parse through
  - They are often simpler to code and intuitive to understand
- But if George was a computer program then this algorithm is not a reasonable strategy to type out “To be or not to be” let alone all of Hamlet, not to mention all the other Shakespeare plays
  - 37 plays with a grand total of 835,997 words!
- This is where a genetic algorithm can be helpful
  - Test the algorithms output with known answer and that will guide us to adjusting the algorithm to getting the correct answer
- We will build an algorithm that can “evolve” to one that can type out Shakespeare plays on its own
- Let’s call our algorithm George to give the poor typing monkey a needed break



https://www.datamation.com/news/tech-comics-brute-force-cyberattack-explained-2.html

BUILDING A GENETIC ALGORITHM TO WRITE SHAKESPEARE

Step 1

Build a Population

Step 2

Selection

- Evaluate Fitness
- Create Mating Pool

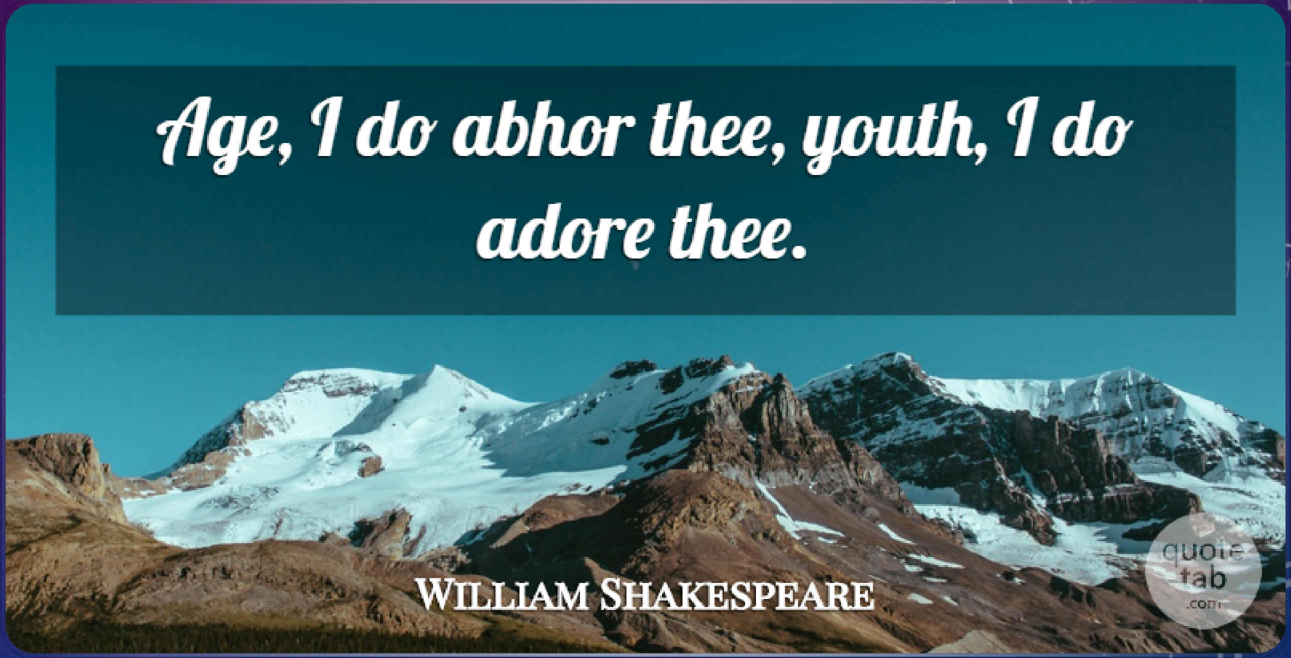
Step 3

Reproduction

- Cross Over
- Mutation

BUILD A POPULATION

- First step is to generate a population of phrases
- Phases are just sequences of strings
- Let’s say we want to build the phrase “Abhor”
    - Phrase is 5 characters long
- We build a population of phases 5 characters long
    - Meows
    - Thigg
    - Astbe
- These phrases have variety in them, however, are still not close enough to “Abhor”



https://www.quotetab.com/quote/by-william-shakespeare/age-i-do-abhor-thee-youth-i-do-adore-thee?source=age

CREATE A POPULATION

- We need a much larger population of randomly generated letters
  - The larger the population, the closer we can get to Abhor
  - With a larger population you are guaranteed phrases that start with “A” and have “b” in second position and “h” in third position and so on
- Larger population of solution set will have enough variety for us to wind down to the correct solution
- Our population will be created with randomly generated elements
- In this case our elements are letters
- In evolution, elements are DNA
- In designing our genetic algorithm, we need to specify our genotypes and phenotypes

GENOTYPE AND PHENOTYPE

Evolution

- Genotype is the genetic code of an organism
- Phenotype is the physical characteristics encoded by genes

Programming

- Genotype is the data structure storing objects properties
- Phenotype is what the variables express

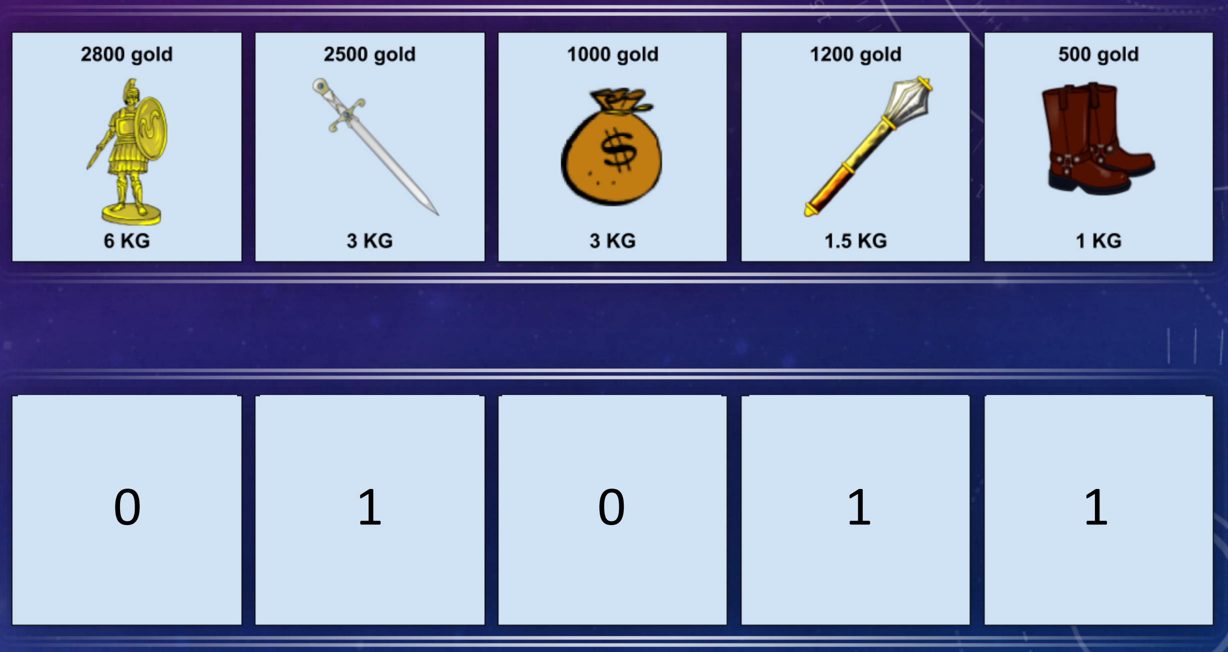
GENOTYPE REPRESENTATION

- An important step is deciding how to represent the solution set for a genetic algorithm
- In the case of George, the genotype and phenotype are both strings of characters
- But in other cases, it can be more complicated
- In the Knapsack problem, you want to know which items you will bring with you on a trip that maximizes the value you will receive for each item while minimizing the weight in the knapsack
    - How do we represent this?

https://heidi-newton.com/blog/introducing-the-knapsack-problem-in-a-classroom

GENOTYPE REPRESENTATION

- In the knapsack problem you can generate the solution set as an array of N items
  - N is the total number of items
- Each index position in the array has a 0 or 1 which represents if the item will be left behind or put in the knapsack
- Here the genotype is 0 or 1
- The phenotype is if the item will be left behind or be put in the knapsack



STEP ONE OF GEORGE THE GENETIC ALGORITHM

- Generate a population of N elements consisting of randomly generated DNA
    - DNA in our case is alphabet letters
    - This will result in a list of phrases sized N

STEP 2: SELECTION

First, we will determine the fitness of each member in the population we generated

Then we will create a mating pool of fit members

FITNESS

- To evaluate fitness, we will use a fitness function
  - Fitness functions assign the program a numeric score reflecting how well it served its purpose
- Members of the population that have a better fitness numerical score are designated as good candidates for our mating pool
- This is akin to Survival of the Fittest in evolution
  - However, in evolution organisms aren't given a score
  - They either survive and pass on their genes to their offspring or they don't

https://www.dzu-doodles.com/

FITNESS FUNCTION

- For the population generated to generate the phrase “Abhor” we will determine fitness by counting the number of correct characters each word has divided by the total number of characters in each phrase.
- This will give a percentage of fitness

Population	Fitness Score
Meows	0%
Thigg	20%
Astbe	40%

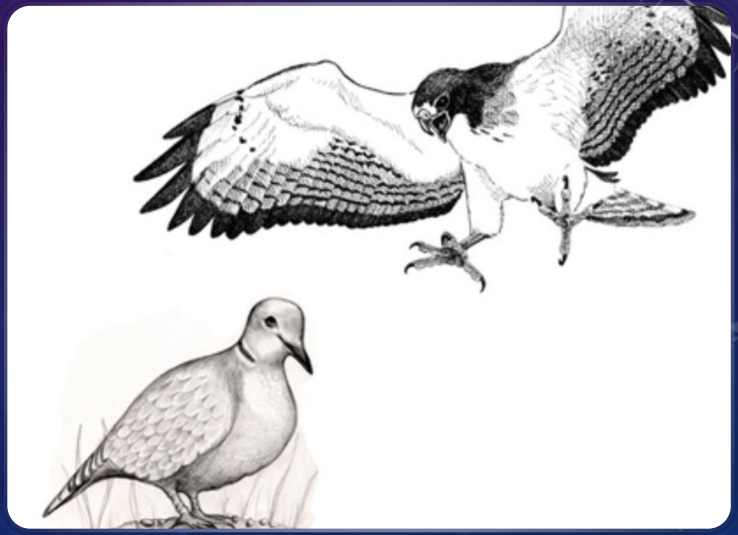
FITNESS FUNCTION

- The fitness function that is picked is very important
- Another fitness function could be how many characters are correct and in the correct place?
- In which case the fitness score will be very different.
- This fitness function intuitively seems as it will give potential candidates for the mating pool
- But this will result with fewer potential candidates with less variety.
- And just as with evolution, the more intuitive features for survival of the fittest are not always the ones that survive.

Population	Fitness Score
Meows	0%
Thigg	0%
Astbe	20%

SURVIVAL OF THE FITTEST

- In evolution, survival of the fittest doesn’t always necessarily mean all organisms with “dominant” traits will beat out all organisms with “submissive” traits.
- This is known as Evolutionary Stable Strategy
- In Richard Dawkin’s book *The Selfish Gene,* he demonstrates this concept by using hawks and doves and game theory



https://medium.com/@mike_lux99/listening-to-chapter-5-of-the-selfish-gene-is-somewhat-chilling-in-light-of-the-recent-election-3d5691b9182b

SURVIVAL OF FITTEST: WHAT IS BETTER A HAWK OR A DOVE?

- We have a species of organisms; one is a hawk the other a dove
- Hawks fight hard and only retreat when injured
- Doves merely threaten in a dignified way and never injure anyone
- If a hawk fights a dove, then the dove runs away
- If a dove meets a dove, then they go on flapping their wings at each other until one gets tired and flies away
- If a hawk meets a hawk, then they fight until one is seriously injured or dies

GAME THEORY: HAWKS AND DOVES

- The survival of the fittest can be expressed in numeric terms in game theory
  - This is just a theory, in real life organisms aren't assigned a fitness numerical score
  - In most organisms, fights don't end with a numerical score
- 50 points are given for a win
- 0 points are rewarded for losing
- -100 points are rewarded for being seriously injured
- -10 points are rewarded for wasting time for a long contest
- Points rewarded can be seen as which species is the "fittest" and are a measure of which species will be surviving in the environment

GAME THEORY: HAWKS AND DOVES

- Our population starts with all doves
- When doves fight one is a winner and receives 50 points, but they both get -10 points for flapping their wings at each other for a long time
  - Average pay off is +15
- Now we have a mutated gene in the population which creates a hawk
- When hawks fight doves, the hawks always win
  - Average pay off is +50 for hawks and -100 for doves
- Average pay off for hawks is higher! Hawk genes run rampant in the population
  - Creating more hawks as they pass on their physical traits to their offspring through their genes
- Hawks are the fittest! They survive!

GO HAWKS*

- After hawks become the dominant species, they now can only fight other hawks
- When a hawk fights another hawk, one always wins and the other always loses
  - Average pay off is now -25
- A single dove in a population of hawks will never win, but also never lose, because they always fly away
  - Average pay off for the dove is now 0
- Dove genes now gain traction and grow in the population

https://www.facebook.com/kcci8/?tn-str=k*F

* The author of this presentation is a proud alumni of the University of Iowa

EVOLUTIONARY STABLE STRATEGY

- Where the hawk strategy seems to be the true Survival of the Fittest, it isn't an evolutionary stable strategy
- Hawk genes and aggressive phenotype will never be the end of the organism's evolution
- Eventually the ratio of hawks and doves will stabilize
- Eventually, hawks and doves will come together and mate
- This will produce offspring that are part hawk and part dove
    - Perhaps they fly away when faced with an enemy they deem will win and they fight when faced with an enemy they deem they will beat
    - The offspring will have superior traits of their ancestors

https://www.richarddawkins.net/2015/07/evolutionarily-stable-strategies-ft-richard-dawkins/

BACK TO OUR FITNESS FUNCTION

As with the hawks and doves, the ultimate fitness function shouldn’t always be one that stops variety in the population

We need a fitness function to separate bad candidates from good ones

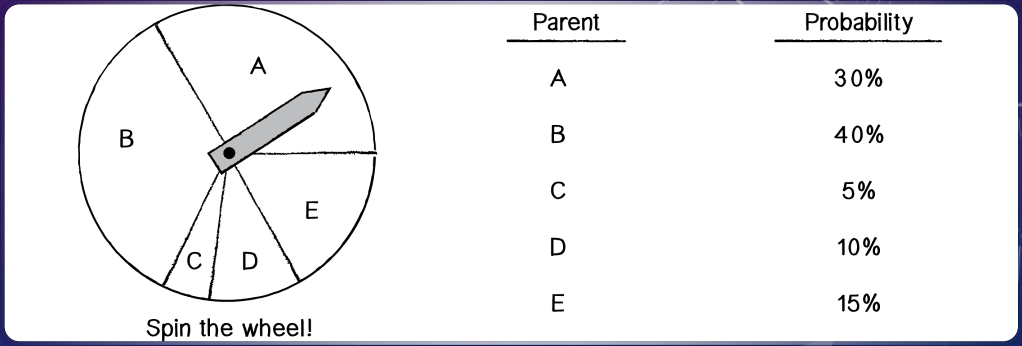
But we also want our fitness function to not filter out so many members in the population so there is no more variety in the population

NOW WE CAN TAKE OUR MEMBERS FROM THE POPULATION THAT ARE FIT AND PLACE THEM INTO OUR MATING POOL

- Two methods to determine the mating pool
  - Elitist
    - Who scored the highest? You are my new Adam and Eve!
    - This however reduces variety
    - With less variety you are now stunting evolution
  - Probabilistic
    - Take the elements and give them a probability score of being the best parent
    - Those with the highest fitness score will have a higher probability of being picked as a parent
    - There's still a chance the lower fitness score members will be picked as a parent
    - This reduces minimizing variety in the resulting population

WHEEL OF FORTUNE

- The probabilistic method is akin to the wheel of fortune
- Say you have 5 members of the population with fitness scores
- Normalize the fitness scores by determining the amount of fitness for each member divided by the total fitness for all members,
  - Then we can express the fitness score as a percentage of fitness in the population
- Now we spin the wheel to determine the members that will be parents in the future generation
- We will select those with the higher fitness score a higher percentage of the time
- While still allowing those with smaller fitness scores to have a chance to populate
  - This way we don't automatically reduce variety in the new population



The Nature of Code figure 9.1

STEP TWO OF GEORGE THE GENETIC ALGORITHM

- Determine the fitness score of each member of the population
- Using a probabilistic method we determine who is mostly likely to be the best parents for the next generation of solutions

STEP 3: REPRODUCTION

Next, we will create children from the parents in our mating pool

Perform Cross Over

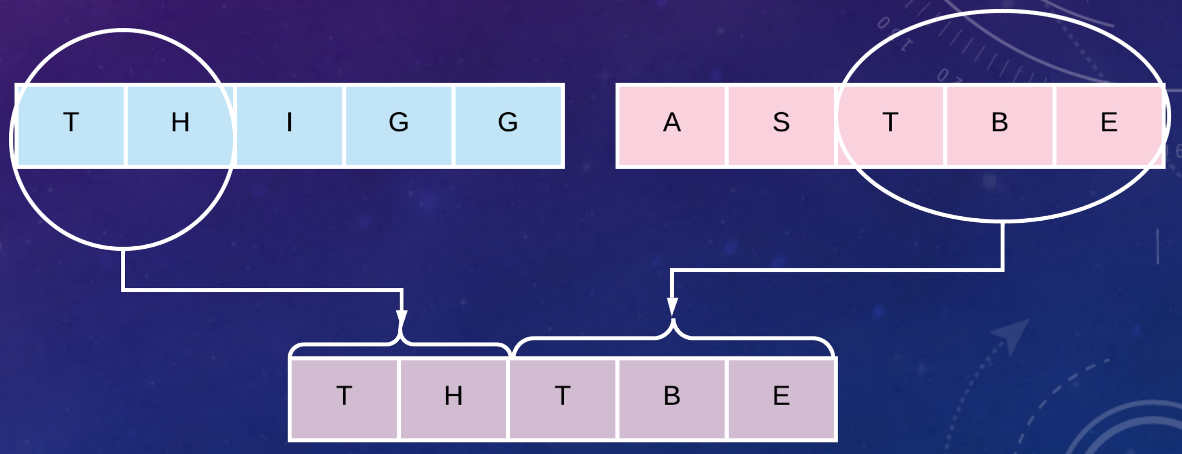
- DNA from each parent is combined into the child
- In the case of George, we will combine letters from each parent

Then Mutation

- This isn't always necessary but does add more variety into the population
- Mutation involves changing the DNA of the child
- In the case of George, letters can be randomly changed

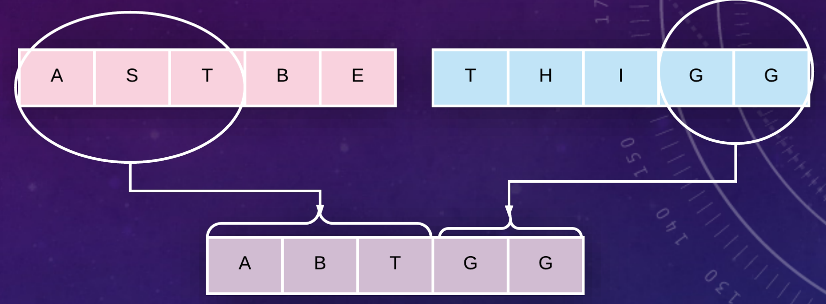
CROSS OVER

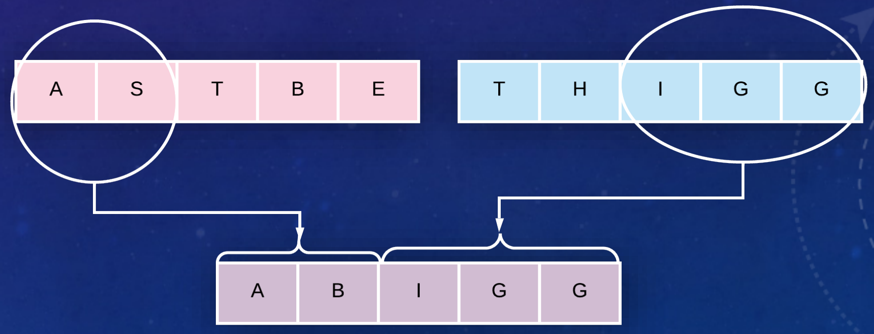
- Back to the earlier example, take 2 parents with a decent fitness score
  - Thigg
  - Astbe
- We're trying to build the word "Abhor"
- We can take parts of Thigg and parts of Astbe to create a child
  - Thtbe
- Still far from our goal however



CROSS OVER

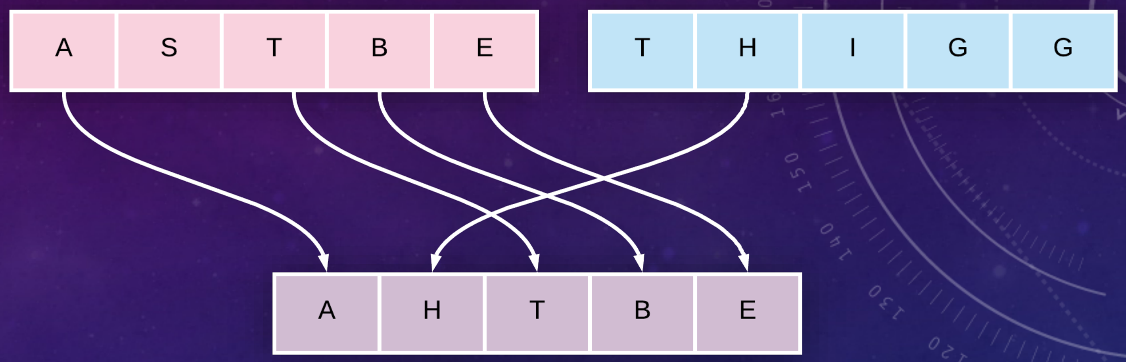
- There are many ways cross over could occur
- The method depends on what the phenotype of the genetic code is
- if the order of the DNA has a large effect on phenotype then the cross over pattern is very important to select
- For George, we are still trying to generate random strings that are closer to the truth, order of the DNA isn't as important.
    - This is again where choice of fitness function arises
    - If our fitness function measured correct characters in the correct placement instead of correct characters, then order of the parent's DNA into the child would be extremely important to preserve!

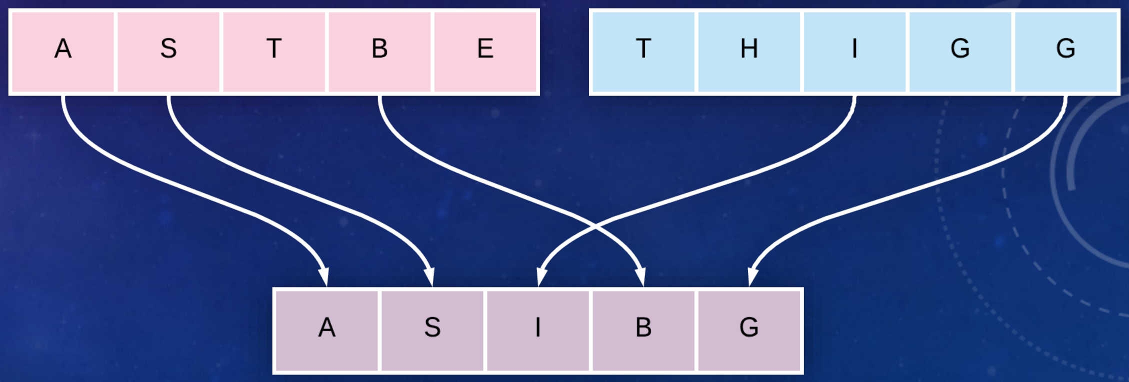




CROSS OVER

- Cross over could also be random
- For each DNA spot (letter position) of the child you randomly select if you will take the mother or fathers DNA (letter)
- Again, it depends on the specific problem you are trying to solve to use random or predefined order of crossover



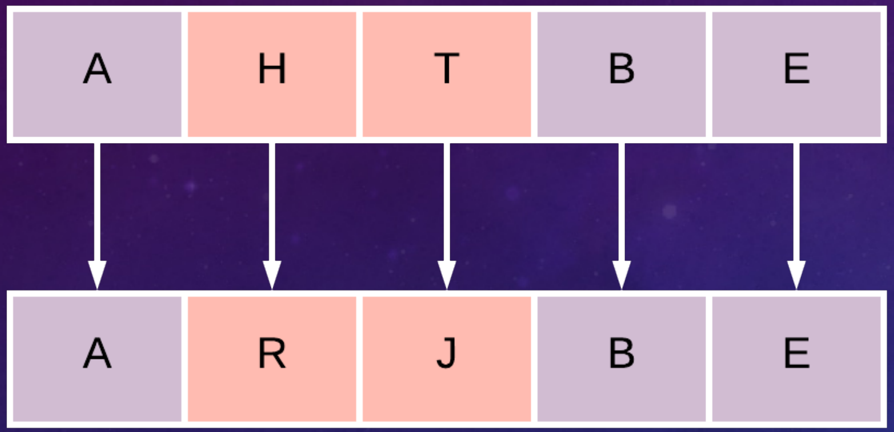


MUTATION

- Mutation is an optional step in a genetic algorithm
- The idea is to add more variety into the population
- The initial population created by George was a randomly generated population
- But with filtering out fitness functions and mating parents, our population's variety might have diminished
- Mutation can introduce variety back into the population to ensure that we can continue to evolve and grow
- Mutation will take an instance of the DNA (a single letter) and change it (randomly select a new letter)

https://www.dreamstime.com/illustration/mutation-cartoon.html

MUTATION



- Mutation is based on a population rate
- If the mutation rate is 40% then 2 of our letters will randomly change
- A high mutation rate isn't always the best strategy
  - If you are randomly selecting new letters for each letter position then you will be stunting evolution
- Which each new generation, the children will end up as randomly generated new members of the population
  - You lose what you gained from checking the parents in the fitness function

STEP THREE OF GEORGE THE GENETIC ALGORITHM

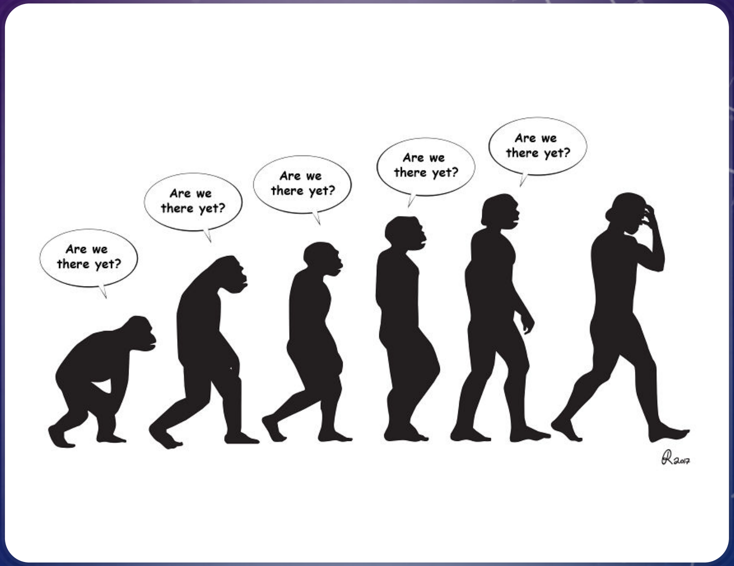
- Take two parents from step 2 and have they reproduce creating a child
- Mutate the child's DNA based on a defined mutation rate
- Add child to new population

GEORGE THE ALGORITHM

- Now we have all the pieces, we can finally build our algorithm
- We start by initializing a population of N randomly generated elements of DNA
- Determine the fitness function for each member of the population select top fit candidates for mating
- Cross over parent DNA to create Child members
- Mutate child members DNA
- Add child members to new population
- Repeat

GEORGE THE ALGORITHM

- Each repetition creates a new population that is “fitter”
  - Each new population will be bringing you closer and closer to matching the correct phrase
- Finally, at the end population you will be able to reproduce a full Shakespearean play



https://www.boredpanda.com/viva-la-evolucion-funny-evolution-silhouette-cartoons-rusty-yunusoff/?utm_source=google&utm_medium=organic&utm_campaign=organic

GENETIC ALGORITHMS

The beauty of genetic algorithms is that they have a wide number of applications

They can be used for artificial creativity

like George!

They can also be used in financial modeling, bioinformatics, social sciences, earth sciences, mathematics and many other fields

The structure of a genetic algorithm is flexible, a genotype and phenotype can be made for any problem an algorithm is faced to solve

ARE GENETIC ALGORITHMS THE MASTER ALGORITHM?

- Genetic algorithms learn the same way that nature does
  - This doesn’t always fit every single problem
- Errors in algorithms can cost lives
  - Such as algorithms tasked with diagnosing cancer or finding the correct drug for a patient
  - In these cases, a genetic algorithm can’t always be trusted to eventually arrive at the correct solution
- Some problems have solution sets so vast that it would take as long as evolution did to generate today’s version of humans to find the correct answer
- And still there might be a better answer
  - Species are continuously evolving
  - Genetic algorithms could also be constantly evolving with each new repetition growing more accurate
- Genetic algorithms can apply to a wide variety of fields
- In some cases, other methods will be needed to solve certain problems

REFERENCES

Domingos, Pedro. *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*. Basic Books, a Member of the Perseus Book Group, 2018.

Dawkins, Clinton Richard. *The Selfish Gene*. Oxford University Press, 2016.

Shiffman, Daniel. *The Nature of Code*. D. Shiffman, 2012.

"A Beginner's Guide to Genetic & Evolutionary Algorithms." *Pathmind*, pathmind.com/wiki/evolutionary-genetic-algorithm.

"Shakespeare's Plays, Listed by Number of Words." *Complete List of Shakespeare's Plays, by Number of Words :|: Open Source Shakespeare*, www.opensourceshakespeare.org/views/plays/plays_numwords.php.