

Abstract

The key concepts of this program were search and computational complexity. This program was a continuation of Project 5 and the elephant simulation program. What this program is meant to do is from Project 5, determine the optimal darting probability of the elephants. How this works is to first use binary search to figure out the different possibilities through trial and error or optimization as the program gets closer to an answer.

Methods

In this project, one place I used computational thinking was to find the best darting probability for controlling elephant populations. I used the binary search strategy concept that we learned in class. It is similar to an adaptive trial and error method where it repeatedly adjusts its guess for the optimal darting probability based on whether the previous guess was too high or too low which narrows the range down and converges to the best solution. This computational thinking allowed me to search the possible darting probabilities.

Results

Reflection

This project could be used to address challenges related to conservation strategies and population control. It helped me better understand how mathematical models and computational thinking can contribute to wildlife conservation efforts. I think similar, but much more advanced and complex programs could be used for broader applications of optimization methods in conservation efforts. This project also showed how to use computational thinking in addressing specific challenges in wildlife preservation.

Follow-up Questions

1. Import statements let you upload libraries into your code so you can use the functions in the file or library.
2. Binary search is a search algorithm which works in an adaptive trial and error way. It finds a value within a range by 'bouncing' around. It is different from linear search because it works by continuously dividing the number range in half until it gets to a target value but linear search checks each number one by one.
3. Binary search is faster than linear search because it halves the search space with each comparison while in a linear search, you would have to go through each value one by one, making it slower.
- 4.
5. I could apply this optimized search algorithm approach to something I'm really interested in by using it to create exercise plans for my physical therapy and knee recovery. I could factor in different things like time or difficulty which would help me personalize the exercises a lot.

Extensions

I didn't do the extension for this project.

References/Acknowledgements

I went to TA hours and got help from Yubin and also got help from Rose. I also used online resources w3schools and stack overflow. I imported libraries sys, elephant, and random.