UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical &
Computer Engineering

ECE 204 *Numerical methods*

Bracketing

Douglas Wilhelm Harder, LEL, M.Math.
dwharder@uwaterloo.ca
dwharder@gmail.com

Introduction

- In this topic, we will
  - Describe the tool of bracketing
  - Review the binary search algorithm
  - Introduce the interpolation search algorithm
  - Discuss the application of bracketing

Bracketing

- We have discussed interpolating polynomials and Taylor series, and both will be used for finding approximations to solutions
  - Sometimes, however, it may not be possible to use such analytic techniques
  - Instead, we may be forced to find an interval on which the solution is known or may possibly exist

Binary search

This reviews a first-year concept,
it is not required for this course.

- As an example of bracketing:
  - Given a sorted array of capacity n, suppose we are searching for the location at which some value is in the array
    - Perhaps to remove it, or store information with the mapping
  - Initially, we only know that the entry exists at some index between 0 and $n - 1$
    - The binary search algorithm says to examine entry $m = \left\lfloor \frac{n}{2} \right\rfloor$
    - If what we are searching for is at that location,
      we're done
    - Otherwise, we have either restricted the search space to one of
      0 to $m - 1$ or $m + 1$ to $n - 1$

Binary search

This reviews a first-year concept, it is not required for this course.

- The efficiency of a binary search is given as O(lg(n)), as the time required to search an array of capacity n is some constant times $\lg(n) = \log_2(n)$
  - This may seem good, as searching an array of capacity one million requires one to check at most 20 entries
  - Question: you are searching for the last name “Zarnett” in the phone book
    - Would you begin by opening the phone book to the middle and then performing a binary search?

Interpolation search

This exemplifies an idea,
it is not required for this course.

- Suppose you are searching a sorted array for the value v and you are currently restricted to viewing entries i through j
  - If v is close to a_i, the next place to examine should be close to i
  - If v is close to a_j, the next place to examine should be close to j
  - If v is approximately half-way between a_i and a_j,

    the next place to examine should be close to $\frac{i+j}{2}$

- The following formula works:

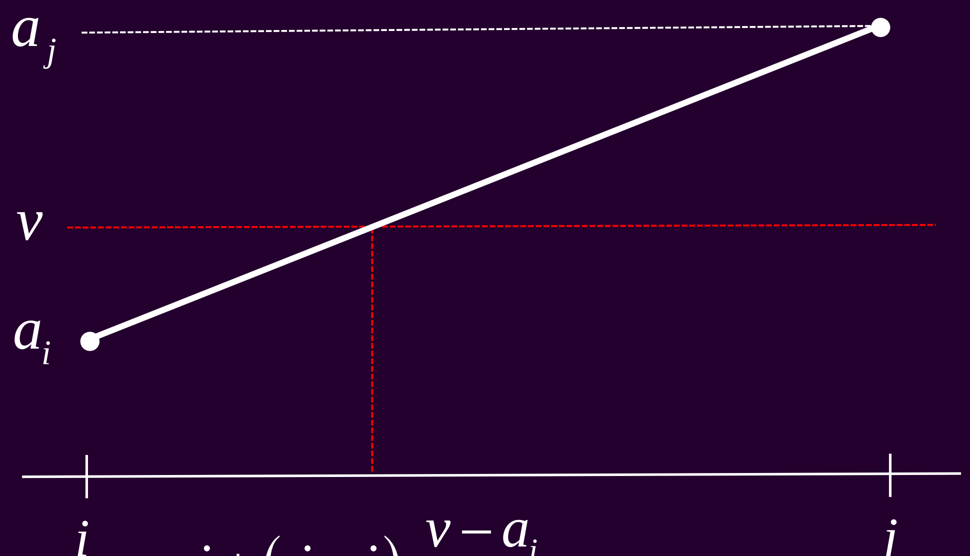
$$i + (j - i)\frac{v - a_i}{a_j - a_i}$$

  - You must check to make sure this value does not equal either i or j and this only works if the values in the array are uniformly distributed

Interpolation search

This exemplifies an idea,
it is not required for this course.

- By the way, this image shows why it is called an interpolation search:



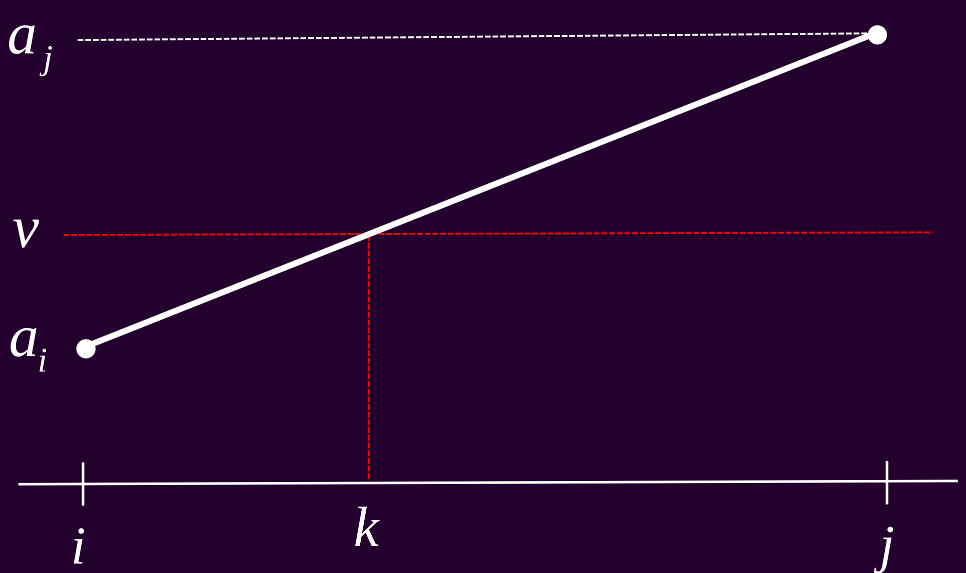
$$i + (j - i)\frac{v - a_i}{a_j - a_i}$$

  - If the array entries are approximately uniformly distributed, the runtime now drops to O(ln(ln(n))

Interpolation search

This exemplifies an idea, it is not required for this course.

- Derivation:



Slope

$$v = a_i + \frac{a_j - a_i}{j - i}(k - i)$$

$$v - a_i = \frac{a_j - a_i}{j - i}(k - i)$$

$$\frac{j - i}{a_j - a_i}(v - a_i) = k - i$$

$$k = i + \frac{j - i}{a_j - a_i}(v - a_i)$$

$$= i + \frac{v - a_i}{a_j - a_i}(j - i)$$

The ratio $\frac{v - a_i}{a_j - a_i}$ gives the proportion v is into the interval $[a_i, a_j]$

Bracketing

- While bracketing may be one of the most efficient algorithm techniques for searching arrays, it is often the least efficient algorithm we can use to approximate solutions numerically
  - Often, we will resort to bracketing only when it is impossible to use the tools of interpolation or Taylor series

Summary

- Following this topic, you now
  - Understand the idea behind bracketing
    - Given that the solution is known to be on an interval, we apply an algorithm to shrink that interval
  - Have reviewed the algorithm of a binary search
  - Understand the idea behind the interpolation search
  - Are aware that bracketing, however, is less optimal for approximation solutions to analytic problems

References

[1] https://en.wikipedia.org/wiki/Binary_search_algorithm
[2] https://en.wikipedia.org/wiki/Interpolation_search

Acknowledgments

None so far.

Colophon

These slides were prepared using the Cambria typeface. Mathematical equations use Times New Roman, and source code is presented using `Consolas`.
Mathematical equations are prepared in MathType by Design Science, Inc.
Examples may be formulated and checked using Maple by Maplesoft, Inc.

The photographs of flowers and a monarch butter appearing on the title slide and accenting the top of each other slide were taken at the Royal Botanical Gardens in October of 2017 by Douglas Wilhelm Harder. Please see

https://www.rbg.ca/

for more information.

Disclaimer