

UNIVERSITY of WISCONSIN-MADISON
Computer Sciences Department

CS 202: Introduction to Computation Professor Andrea Arpaci-Dusseau

How can computation... find what you're looking for - faster?

© Original Artist: Steve Meyers
Reproduction rights obtainable from www.CartoonStock.com

Motivating Exercise

Play 20 questions in pairs

- Repeat few times
 - Person A thinks of a number between 1 and 100
 - Person B guesses number with YES/NO questions
 - Try to ask as few questions as possible!
 - Record how many guesses needed
- Switch roles and play few times more

Repeat with numbers between 1 and 1000

- Handout sheet of number grid may be useful
- Cross off guesses or numbers that secret can't be

How many guesses on average did it take you?

Use Number Grid to Track Guesses

Best algorithm for searching?

BINARY SEARCH

Guess number midway between "lo" and "hi"
 (lo starts out at 1, hi at 1000, midway = 500)

Ask "Lower than this midway number?"

If Yes then

- Set hi = midway - 1
- Guess number 1/2 btwn lo and hi (< 250?)

ELSE

- Set lo = midway
- Guess number 1/2 btwn lo and hi (< 750?)

Repeat

Play guessing game again with 1000 numbers - should need 10 or fewer guesses!

How would you implement Binary Search for Key?

Exercise Guessing Game:

What is the secret to be guessed?

- Integer between 1 and 1000 partner is thinking of
- Ask if guess is lower (or higher) than secret

Binary Search for Specified Key:

What is the secret to be guessed?

- Secret is **index in List** holding key we are looking for
- Guess position in list
- Ask if item at guess is lower (or higher) than key

Different Assumptions for Linear vs. Binary Search?

Binary search assumes list is sorted!

- Does not work unless items in list are in order

Trade-off: For a fast search, should application spend time to sort data or not?

- Will look at sorting algorithms later...

Review: Linear Search

```

when clicked
  broadcast Create Valuable Numbers and wait
  ask What number should I search for? and wait
  set Key_ to answer
  broadcast Find Key_ and wait
  if not Key Index = 0
    say join The key Key_ is located at index Key Index for 2 secs
  else
    say join The matching name of item Key Index of Valuable Numbers_
  say join join The key Key_ was not found!

when I receive Find Key_
  set Index_ to 1
  set Key Index_ to 0
  repeat length of Valuable Numbers_
    if item Index_ of Valuable Numbers_ = Key_
      set Key Index_ to Index_
      stop script
  change Index_ by 1
    
```

Algorithm checks every element in list (in order) to see if it is the one...

Variables

- Valuable Numbers List
- Key: Input - What we are searching for
- Key Index: Output - Index where we found Key
- index: local variable

Binary Search in Scratch

Goal: Same inputs and outputs to script, but faster

```

when I receive Binary Search_
  set Key Index_ to 0
  set hi_ to 1000
  set lo_ to 1
  repeat until lo > hi
    set index_ to round hi + lo / 2
    if item index_ of Valuable Numbers_ = Key_
      set Key Index_ to index_
      stop script
    else if item lo of Valuable Numbers_ > Key_
      set lo to index_ + 1
    else
      set hi to index_ + 1
    
```

Same Variables

- Valuable Numbers List
- Key: Input - What we are searching for
- Key Index: Output - Index where we found Key
- index: local variable

Faster: Use index to skip around List efficiently

Invariant (condition always holds true)

- $lo \leq$ Index of Secret key $\leq hi$
- True before loop begins
- True every time after

Running Binary Search: Ex 1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
10	15	28	31	32	33	49	60	62	68	69	74	81	85	96	97

```

when I receive Binary Search
  set Key Index to 0
  set hi to length of Valuable Numbers
  set lo to 1
  repeat until lo > hi
    set index to round (hi + lo / 2)
    if Item Index of Valuable Numbers = Key
      set Key Index to index
      stop script
    if Item Index of Valuable Numbers > Key
      set hi to index - 1
    else
      set lo to index + 1
  
```

Looking for key 85

Loop	index	Item	Item > Lo	Hi	Key?
Initial					
1					
2					
3					
4					

Round: rounds up to nearest integer
(hint: round(8.5) = 9)

Running Binary Search: Ex 2

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
10	15	28	31	32	33	49	60	62	68	69	74	81	85	96	97

```

when I receive Binary Search
  set Key Index to 0
  set hi to length of Valuable Numbers
  set lo to 1
  repeat until lo > hi
    set index to round (hi + lo / 2)
    if Item Index of Valuable Numbers = Key
      set Key Index to index
      stop script
    if Item Index of Valuable Numbers > Key
      set hi to index - 1
    else
      set lo to index + 1
  
```

Looking for key 33

Loop	index	Item	Item > Lo	Hi	Key?
Initial					
1					
2					
3					
4					

Running Binary Search: Ex 3

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
10	15	28	31	32	49	60	62	68	69	74	81	85	96	97	

```

when I receive Binary Search
  set Key Index to 0
  set hi to length of Valuable Numbers
  set lo to 1
  repeat until lo > hi
    set index to round (hi + lo / 2)
    if Item Index of Valuable Numbers = Key
      set Key Index to index
      stop script
    if Item Index of Valuable Numbers > Key
      set hi to index - 1
    else
      set lo to index + 1
  
```

Looking for key 34

Loop	index	Item	Item > Lo	Hi	Key?
Initial					
1					
2					
3					
4					

What does Key Index equal when script ends?
Key Index still 0
Can use to signify that key not in List

How many guesses to find Key with Linear Search?

```

when clicked
  broadcast Find Key and wait
  set Key to answer
  broadcast Find Key and wait
  if not Key Index = 0
    say join join The key Key join is located at index Key Index for 2 secs
  else
    say join join The key Key join was not found

when I receive Find Key
  set index to 1
  set Key Index to 0
  repeat length of Valuable Numbers
    if Item Index of Valuable Numbers = Key
      set Key Index to index
      stop script
  change index by 1
  
```

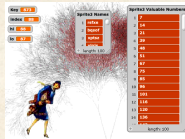
How many loops?

N = Elements in List

- Best case (minimum)?
- 1 loop!
- Worst case (maximum)?
- N loops
- Average case?
- N/2 loops
- O(N)
- Just like Find Max

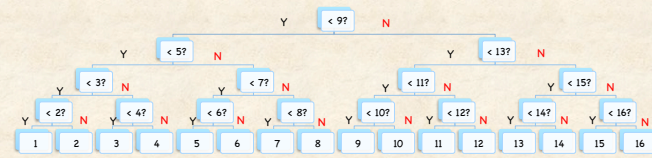
How many Guesses Needed?

N	16	32	64	128	256	512	1024	2048	4096
Linear									
Binary									



Decision Tree for 1..16

How many questions needed to find answer between 1 and 16 (16 numbers)?



Only 4 questions for 16 numbers

Decision Tree for 1..32

How many questions needed to find answer between 1 and 32 (32 numbers)?
Double range of numbers covered



Just one more question --> 5 questions

Decision Tree for 1..64

How many questions needed to find answer between 1 and 64 (64 numbers)?
Double again range of numbers covered



6 questions

Decision Tree for N items

How many questions needed for N integers?



Each level of tree corresponds to one question
How deep (or high) is tree of N integers?

- 1 question → 2 numbers
- 2 questions → 4 numbers
- 3 questions → 8 numbers
- 4 questions → 16 numbers
- 5 questions → 32
- 6 questions → 64
- Q questions → ??

How to go from N to Q?

$$Q = \log_2 N$$

$$2^Q \text{ numbers}$$

Complexity of Binary Search

```

when I receive Binary Search
  set Key Index to 0
  set hi to length of Valuable Numbers
  set lo to 1
  repeat until lo > hi
    set index to round hi + lo / 2
    if item index of Valuable Numbers = Key
      set Key Index to index
      stop script
    if item index of Valuable Numbers > Key
      set hi to index - 1
    else
      set lo to index + 1
  
```

How many iterations?

Worst case:
Operations: $O(\log_2 N)$

Modification to decision tree
(2 if's per loop)
Stop early if item at index = key

Game of 20 Questions



How many objects can you choose between with 20 questions?



2^{20}
This is approximately 1 million objects! (1,048,576)

How to be a good player of 20 questions?

Check-Up



Is it possible for index of secret key to be $< lo$?

What assumptions does binary search make?

How many iterations of repeat loop are needed for a list containing 1024 (1K) elements?

1048576 (1024 K) elements?

```

when I receive Binary Search
  set Key Index to 0
  set hi to [ ]
  set lo to [ ]
  repeat until lo > hi
    set index to round hi + lo / 2
    if item index of Valuable Numbers = [ ]
      set Key Index to [ ]
      stop script
    if item index of Valuable Numbers > [ ]
      set hi to index - 1
    else
      set lo to index + 1
  
```

Announcements

How to efficiently search for element in a List

- $O(N)$ guesses to find using Linear Search
- $O(\log_2 N)$ guesses to find using Binary Search (depth of tree)
 - Assumes data is sorted!

Homework 5: Due Friday at 5pm

- Gallery will open Friday morning for Extra Credit submissions; submit by Friday midnight
- Vote over weekend for Round 1
- Can you figure out how Mozart Dice Game works?