Clicker question

• What did you think of the sandwich building exercise?

(A) Very interesting – I learned a lot
(B) Interesting – I learned something
(C) Not so sure – maybe there are more effective ways to get those points across
(D) Didn't like it too much – not really worth the time we spent on it
(E) What a waste of time!

Selected RQs

"Does making algorithms sometimes make a simple operation more complicated because, while it may be easy to complete on paper, it is actually difficult to code?"

(Rebecca)

Textbook, Ch.10/5

Common problem (identified by quite a few students):
“I didn’t understand the IAL/NAL algorithms”

Let's work through this together ...

Input: $n$ lists of terms (e.g., from an index of web pages)

Output: list of all terms that occur in all input lists

Intersect Alphabetized List

Assumption:
All input lists are in alphabetic (= lexicographic) order

Procedure Intersect Alphabetized Lists:
1. Put a marker/arrow at the start of each list
2. If all markers point to the same item, save it, because they are the same item
3. Move the marker(s) to the next position for whichever item is the earliest in the alphabet
4. Repeat Step 2-3 until some marker reaches the end of the list
Intersect Alphabetized Lists

List 1
Elephants Have Humongous Noses Today

List 2
Amy's Elephants Have No Noses

List 3
Elephants Fix Flying Noses In The Zoo

Intersect List
Elephants

Intersect Alphabetized Lists

List 1
Elephants Have Humongous Noses Today

List 2
Amy's Elephants Have No Noses In The Zoo

List 3
Elephants Fix Flying Noses

Intersect List
Elephants

Intersect Alphabetized Lists

List 1
Elephants Have Humongous Noses Today

List 2
Amy's Elephants Have No Noses In The Zoo

List 3
Elephants Fix Flying Noses

Intersect List
Elephants

Intersect Alphabetized Lists

List 1
Elephants Have Humongous Noses Today

List 2
Amy's Elephants Have No Noses In The Zoo

List 3
Elephants Fix Flying Noses

Intersect List
Elephants

Intersect Alphabetized Lists

List 1
Elephants Have Humongous Noses Today

List 2
Amy's Elephants Have No Noses In The Zoo

List 3
Elephants Fix Flying Noses

Intersect List
Elephants

Intersect Alphabetized Lists

List 1
Elephants Have Humongous Noses Today

List 2
Amy's Elephants Have No Noses In The Zoo

List 3
Elephants Fix Flying Noses

Intersect List
Elephants

Intersect Alphabetized Lists

List 1
Elephants Have Humongous Noses Today

List 2
Amy's Elephants Have No Noses In The Zoo

List 3
Elephants Fix Flying Noses

Intersect List
Elephants

Intersect Alphabetized Lists

List 1
Elephants Have Humongous Noses Today

List 2
Amy's Elephants Have No Noses In The Zoo

List 3
Elephants Fix Flying Noses

Intersect List
Elephants

Intersect Alphabetized Lists

List 1
Elephants Have Humongous Noses Today

List 2
Amy's Elephants Have No Noses In The Zoo

List 3
Elephants Fix Flying Noses

Intersect List
Elephants
Intersect Alphabetized Lists

List 1
Elephants
Have
Humongous
Noses
Today

List 2
Amy's
Elephants
Have
No
Noses

List 3
Elephants
Fix
Flying
Noses
In
The
Zoo

Intersect List
Elephants
Noses

MATCH!!

Intersect Alphabetized Lists

List 1
Elephants
Have
Humongous
Noses
Today

List 2
Amy's
Elephants
Have
No
Noses

List 3
Elephants
Fix
Flying
Noses
In
The
Zoo

Intersect List
Elephants
Noses

Intersect Alphabetized Lists

List 1
Elephants
Have
Humongous
Noses
Today

List 2
Amy's
Elephants
Have
No
Noses

List 3
Elephants
Fix
Flying
Noses
In
The
Zoo

Intersect List
Elephants
Noses

Intersect Alphabetized Lists

List 1
Elephants
Have
Humongous
Noses
Today

List 2
Amy's
Elephants
Have
No
Noses

List 3
Elephants
Fix
Flying
Noses
In
The
Zoo

Intersect List
Elephants
Noses

MATCH!!
Intersect Alphabetized Lists

Assumption: Input lists are *not* in lexicographic order

Procedure *No Alphabetized Lists*:
1. Put a marker/arrow at the start of each list
2. If markers point to the same item, save it
3. Move marker of one list down by one
4. If the marker is at the end of the list, move the marker to the top and advance the marker on the next list by one
5. Repeat 2-4 until all markers are at the end of the list
No Alphabetized Lists

List 1
Socks
Box
Knox

List 2
Socks
On
Knox

List 3
Knox
In
Box

Intersect List

List 1
Socks
Box
Knox

List 2
Socks
On
Knox

List 3
Knox
In
Box

Intersect List

Connecting with Computer Science | cs.ubc.ca/~hoos/cpsc101

31

32

33

34
No Alphabetized Lists

List 1: Socks
List 2: Socks
List 3: Knox

Intersect List: Socks

List 1: Box
List 2: On
List 3: Knox

Intersect List: Knox

List 1: Socks
List 2: Box
List 3: Knox

Intersect List: Knox

List 1: Socks
List 2: On
List 3: Knox

Intersect List: Knox

List 1: Socks
List 2: Box
List 3: Knox

Intersect List: Knox
No Alphabetized Lists

List 1
Socks
Box
Knox

List 2
Socks
On
Knox

List 3
Knox
In
Box

Intersect List

MATCH!!

No Alphabetized Lists

List 1
Socks
Box
Knox

List 2
Socks
On
Knox

List 3
Knox
In
Box

Intersect List

No Alphabetized Lists

List 1
Socks
Box
Knox

List 2
Socks
On
Knox

List 3
Knox
In
Box

Intersect List

No Alphabetized Lists

List 1
Socks
Box
Knox

List 2
Socks
On
Knox

List 3
Knox
In
Box

Intersect List

MATCH!!
No Alphabetized Lists

List 1
- Socks
- Box
- Knox

List 2
- Socks
- On
- Knox

List 3
- Knox
- In
- Box

Intersect List
- Knox

We are at the end of all the lists so we are done!

IAL vs NAL

- **Intersect Alphabetized List (IAL):**
  - Requires input lists to be in alphabetical order
  - Requires much fewer steps than NAL

- **No Alphabetized List (NAL):**
  - Does not require the lists to be in alphabetical order
  - Requires many more steps than IAL

**Lesson learned:** Organizing data in certain ways can make computations much more efficient!

IAL vs NAL

- Steps in IAL: 5+5+7 = 17
- Steps in NAL: 5*5*7 = 175

A bigger example:
- Steps in IAL: 5+7+7+8+6+6 = 39
- Steps in NAL: 5*7*7*8*6*6 = 70560
Processes: Programs & Algorithms

Processes (programs, algorithms)

- are everywhere
- rule!
- allow us to make computers (and people) do things they otherwise couldn’t

Jumping in at the deep end

Fundamental concepts:
- Instructions (input, output, drawing, math, …)
- Sequences of instructions
- Variables
- Loops / repetition (repeat, until, while, for)
- Procedures / functions
- Conditional statements (if, else)
- Arrays

Don’t Panic!

- It’s OK to not understand parts of a program
- Web search can help
- Experimentation aids understanding

Fundamental concepts:
- Instructions (input, output, drawing, math, …)
- Sequences of instructions
- Variables
- Loops / repetition (repeat, until, while, for)
- Procedures / functions
- Conditional statements (if, else)
- Arrays