ALGORITHMS

David Kauchak
CS 140 – Fall 2022

1

Introductions

Dr. | Prof | Professor

Dave | Kauchak

Pronouns: he/him/his

2

Algorithms

"For me, great algorithms are the poetry of computation. Just like verse, they can be terse, allusive, dense and even mysterious. But once unlocked, they cast a brilliant new light on some aspect of computing." – Francis Sullivan

What is an algorithm?

3

Example algorithms

- sort a list of numbers
- find a route from one place to another (cars, packet routing, phone routing, ...)
- find the longest common substring between two strings
- add two numbers
- microchip wiring/design (VLSI)
- solve sudoku
- cryptography
- compression (file, audio, video)
- spell checking
- pagerank
- classify a web page

4

To the course webpage…

5

6

Pseudocode

A way to discuss how an algorithm works that is language agnostic and without being encumbered with actual implementation details.

Should give enough detail for a person to undersand, analyze and implement the algorithm.

7

Pseudocode examples

```
MYSTERY1(A)
x ← −∞
for i ← 1 to length[A]
        if A[i] > x
                x ← A[i]
return x
```

```
MYSTERY2(A)
for i ← 1 to ⌊length(A)/2⌋
        swap A[i] and A[length(A) − (i − 1)]
```

8

Pseudocode convections

array indices start at 1 not 0

we may use notation such as ∞, which, when translated to code, would be something like Integer.MAX VALUE

use shortcuts for simple function (e.g. swap) to make pseudocode simpler

we'll often use ← instead of = to avoid ambiguity

indentation specifies scope

9

Proofs

What is a proof?

- A deductive argument showing a statement is true based on previous knowledge (axioms)

Why are they important/useful?

- Allows us to be sure that something is true
- In algs: allow us to prove properties of algorithms

10

An example

Prove the sum of two odd integers is even

11

Proof techniques?

example/counterexample

enumeration

by cases

by inference (aka direct proof)

trivially

contrapositive

contradiction

induction (strong and weak)

12

Proof by induction (weak)

Proving something about a sequence of events by:

1. first: proving that some starting case is true and
2. then: proving that if a given event in the sequence were true then the next event would be true

13

Proof by induction (weak)

1. **Base case:** prove some starting case is true
2. **Inductive case:** Assume some event is true and prove the next event is true
   a. **Inductive hypothesis:** Assume the event is true (usually k or k-1)
   b. **Inductive step to prove:** What you're trying to prove *assuming* the inductive hypothesis is true
   c. **Proof of inductive step**

14

Proof by induction example

Prove: $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

1. **Base case:** prove some starting case is true
2. **Inductive case:** Assume some event is true and prove the next event is true
   a. **Inductive hypothesis:** Assume the event is true (usually k or k-1)
   b. **Inductive step to prove:** What you're trying to prove *assuming* the inductive hypothesis is true
   c. **Proof of inductive step**

15

Base case

Prove: $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

Show it is true for n = 1

$$\sum_{i=1}^{n} i = 1 = \frac{1 * 2}{2}$$

16

Inductive case

Prove: $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

Inductive hypothesis: assume n = k – 1 is true

$$\sum_{i=1}^{k-1} i = \frac{(k-1)*k}{2}$$

17

Inductive case

Prove: $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

Inductive hypothesis: assume n = k – 1 is true

$$\sum_{i=1}^{k-1} i = \frac{(k-1)k}{2}$$

Prove:

$$\sum_{i=1}^{k} i = \frac{k(k+1)}{2}$$

18

Inductive case: proof

Prove: $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ IH: $\sum_{i=1}^{k-1} i = \frac{(k-1)k}{2}$

$$\sum_{i=1}^{k} i =$$

$$= \frac{k(k+1)}{2}$$

19

Inductive case: proof

Prove: $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ IH: $\sum_{i=1}^{k-1} i = \frac{(k-1)k}{2}$

$\sum_{i=1}^{k} i = k + \sum_{i=1}^{k-1} i$ by definition of sum

$= k + \frac{(k-1)*k}{2}$ by IH

$= \frac{2k}{2} + \frac{(k-1)*k}{2}$

$= \frac{2k + (k-1)*k}{2}$

$= \frac{k^2+k}{2}$

$= \frac{k(k+1)}{2}$

Why does this work?

20

Sorting

Input: An array of numbers A

Output: The number in sorted order, i.e.,

$$A[i] \leq A[j] \; \forall i < j$$

21

Sorting

What sorting algorithm?

```
for j ← 2 to length[A]
    current ← A[j]
    i ← j − 1
    while i > 0 and A[i] > current
        A[i + 1] ← A[i]
        i ← i − 1
    A[i + 1] ← current
```

22

Sorting

```
INSERTION-SORT(A)
for j ← 2 to length[A]
    current ← A[j]
    i ← j − 1
    while i > 0 and A[i] > current
        A[i + 1] ← A[i]
        i ← i − 1
    A[i + 1] ← current
```

23

```
INSERTION-SORT(A)
for j ← 2 to length[A]
    current ← A[j]
    i ← j − 1
    while i > 0 and A[i] > current
        A[i + 1] ← A[i]
        i ← i − 1
    A[i + 1] ← current
```

Does it terminate?

Is it correct?

How long does it take to run?

Memory usage?

24