csce350 — Data Structures and Algorithms
Fall 2019 — Lecture Notes: Counting Operations in Non-recursive Algorithms

This document contains slides from the lecture, formatted to be suitable for printing or individual reading, and with occasional supplemental explanations added. It is intended as a supplement to, rather than a replacement for, the lectures themselves — you should not expect the notes to be self-contained or complete on their own.

1 Introduction

Now let's "zoom in" on Step 3, in which we need to **count** the number of times the basic operation is executed. 2.3

If the algorithm is a non-recursive algorithm, then we can usually

WRITE AND SOLVE A SUMMATION.

2 Example: MaxElement

```
MAXELEMENT(A[0, ..., n − 1])
  m ← A[0]
  for i ← 1, ..., n − 1 do
    if A[i] < m then
      m ← A[i]
    end if
  end for
  return m
```

3 Analysis for MaxElement

$$\begin{aligned} C(n) &= \sum_{i=1}^{n-1} \\ &= (n-1) - 0 + 1 \\ &= n - 1 \\ &\in \Theta(n) \end{aligned}$$

4 *Example: ElementsUnique*

```
ElementsUnique(A[0, ..., n − 1])
  for i ← 0, ..., n − 2 do
    for j ← i + 1, ..., n − 1 do
      if A[i] = A[j] then
        return false
      end if
    end for
  end for
  return true
```

5 *Analysis for ElementsUnique*

$$
\begin{aligned}
C(n) &= \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} 1 \\
&= \sum_{i=0}^{n-2} \big((n-1) - (i+1) + 1\big) \\
&= \sum_{i=0}^{n-2} \big(n - i - 1\big) \\
&= \sum_{i=0}^{n-2} (n-1) - \sum_{i=0}^{n-2} i \\
&= (n-1) \sum_{i=0}^{n-2} 1 - \frac{(n-2)(n-1)}{2} \\
&= (n-1)(n-2-0+1) - \frac{(n-2)(n-1)}{2} \\
&= \cdots = \frac{n(n-1)}{2} \in \Theta(n^2)
\end{aligned}
$$

6 *Example: Matrix multiplication*

See textbook, page 64.

7 *Example: Binary Digits*

```
CountBinaryDigits(n)
  c ← 0
  while n > 1 do
    c ← c + 1
    n ← ⌊n/2⌋
  end while
  return c
```

For this algorithm, there is no direct way to write a summation, because we don't know the number of iterations. But we can write a recurrence, which leads directly to the next section.