Algorithms

CS 231
Dianna Xu

Algorithm

- Step-by-step method for performing some action
- An algorithm is a sequence of finite instructions, often used for calculation (Wikipedia)

Pseudo code

- Variables
  - Memory storage
  - Size and type
- Assignment statement
  - $x := \text{expr}$

Conditional

- $\text{if (condition)} \ \text{then s1 else s2}$

Structure

```plaintext
if (x == 2)
  some statements here
else
  some statements here
```

while loop

```plaintext
while (condition)
  body
end while
```

Condition

```plaintext
Case 1

Case 2
```

while (condition)

process
for loop

\[
\text{for } (i := \text{initial expr to final expr }) \\
\quad \text{body} \\
\text{next } i
\]

Greatest common divisor

- \( a, b \in \mathbb{Z}, a \neq 0, b \neq 0, \gcd(a, b) \) is the integer \( d \) with the following properties:
  - \( d | a \) and \( d | b \)
  - \( \forall c \in \mathbb{Z}, \text{if } c | a \text{ and } c | b, \text{ then } c \leq d \)

Lemmas

- **Lemma 1:**
  - If \( r \) is a positive integer, then \( \gcd(r, 0) = r \)
- **Lemma 2:**
  - Given \( a, b \in \mathbb{Z} \), with \( b \neq 0 \), and \( q, r \in \mathbb{Z} \) such that \( a = bq + r \)
  - \( \text{Then } \gcd(a, b) = \gcd(b, r) \)
    - Show \( \gcd(a, b) \leq \gcd(b, r) \)
    - And \( \gcd(b, r) \leq \gcd(a, b) \)

The Euclidean Algorithm

- By hand: \( \gcd(123, 456) \)
  - \( 456 = 123 \times 3 + 87 \rightarrow \gcd(456, 123) = \gcd(123, 87) \)
  - \( 123 = 87 \times 1 + 36 \rightarrow \gcd(123, 87) = \gcd(87, 36) \)
  - \( 87 = 36 \times 2 + 15 \rightarrow \gcd(87, 36) = \gcd(36, 15) \)
  - \( 36 = 15 \times 2 + 6 \rightarrow \gcd(36, 15) = \gcd(15, 6) \)
  - \( 15 = 6 \times 2 + 3 \rightarrow \gcd(15, 6) = \gcd(6, 3) \)
  - \( 6 = 2 \times 3 \rightarrow \gcd(6, 3) = 3 \)

Algorithm: Euclidean

**Input:** \( A, B \) (\( A, B \) in \( \mathbb{Z} \), \( A > B \geq 0 \))

**Algorithm Body:**

\[
\begin{align*}
  &a := A, \quad b := B, \quad r := B \\
  &\text{while } (b \neq 0) \\
  &\quad r := a \mod b \\
  &\quad a := b \\
  &\quad b := r \\
  &\text{end while} \\
\end{align*}
\]

**Output:** \( \gcd := a \)

Properties of an Algorithm

- Input
- Output
- Definiteness
- Correctness
- Finiteness
- Effectiveness
- Generality
Finding the Max Element

- Give an algorithm for finding the largest value in a finite sequence of integers

**Input:** $a_1, a_2...a_n \in \mathbb{Z}$

**Algorithm Body:**

1. $max := a_1$
2. for $i := 2$ to $n$
   - if ($max < a_i$) then $max := a_i$
3. next $i$

**Output:** $max$

Searching

- Locating an element in an ordered list

- $1 2 3 5 6 7 8 10 12 13 15 16 18 19 20 22$

- Linear Search – Sequential Search

- Binary Search
  - $1 2 3 5 6 7 8 10$
  - $12 13 15 16$
  - $18 19$
  - $18$

- $18$

- $12$

Running Time

Sorting

- Ordering the elements of a list

- Bubble Sort
  - compare each pair and swap if necessary

- Insertion Sort
  - the front of the list is kept in order
  - the sorted list starts with 1 element, the first
  - each successive element is compared and inserted into the correct position in the sorted list

Greedy Algorithm

- Optimization problems – to find a solution to the given problem that either maximizes or minimizes the value of some parameter

- The simplest approach – greedy
  - select the best available choice at each step
  - does not consider consequences of all sequences
  - solution is not always optimal