Overview

• HW6 due on Tuesday
• Last Lecture: Algorithms
  – Big-O notation of Growth of Complexity Function
  – Tractable vs Intractable Problems
  – Computability theory
  – Number theory: Division, Primes, GCD, LCM
• This Lecture: Rest of Algorithms & Graphs
  – Euclidean Algorithm for GCD
  – Modular Arithmetic: Properties of Congruence
  – Introduction of Graphs
  – Graph theory

Application: Program Correctness

Euclidean Algorithm

• By Greek Mathematician Euclid (2300 years ago), one of the oldest known algorithm. Finds the greatest common divisor of 2 nonnegative integers $a$ and $b$

• Example, if we want to find gcd(287, 91), we divide 287 by 91, and get $287 = 91 \times 3 + 14$ \hspace{1cm} // i=j*q+r in program

In the next step, if the remainder is not zero, we divide 91 (divisor) by 14 (remainder) and get $91 = 14 \times 6 + 7$

So again we divide 14 by 7 and get to remainder 0! Then we stop $14 = 7 \times 2 + 0$

Therefore, gcd(287, 91) = 7.
Euclidean Algorithm:

GCD(nonnegative integer a; nonnegative integer b)
//input : a ≥ b, not both a and b are zero

Local variables: integers i, j
i = a; j = b  // initialize value of i and j

while j ≠ 0 do
    compute i = qj + r, 0 ≤ r < j
    i = j
    j = r
end while

//i now has the value gcd(a,b)
return i;
end function GCD

Property * : \((\forall \text{ integers } a,b,q,r)[(a = bq + r) \rightarrow \gcd(a,b) = \gcd(b,r)]\)
Proof by contradiction

Assume that \(a = bq + r\) and \(\gcd(a,b) \neq \gcd(b,r)\)

Let \(a = \gcd(a,b) \cdot c_1\) and \(b = \gcd(a,b) \cdot c_2\) for some integers \(c_1\) and \(c_2\)

\(r = a - qb = \gcd(a,b) \cdot c_1 - q \cdot \gcd(a,b) \cdot c_2 = \gcd(a,b) \cdot (c_1 - qc_2)\)

So \(\gcd(a,b)\) divides \(r\) (also divides \(a\) and \(b\), by definition of \(\gcd(a,b)\))

Similarly we can show that \(\gcd(b,r)\) divides \(a\) (also divide \(b\) and \(r\))

As \(\gcd(a,b) \neq \gcd(b,r)\)

Case 1: \(\gcd(a,b) > \gcd(b,r)\), then \(\gcd(b,r)\) cannot be the gcd of \(b\) and \(r\).
(since \(\gcd(a,b)\) is greater value and divides both \(b\) and \(r\))
Contradiction!

Case 2: \(\gcd(a,b) < \gcd(b,r)\) then \(\gcd(a,b)\) cannot be the gcd of \(a\) and \(b\)
(similar argument as case 1), Contradiction!
• To prove the correctness of this function, we need to prove the following

Let \( i_n \) and \( j_n \): values of \( i \) and \( j \) after the \( n^{th} \) loop in Euclidean Algorithm.

We need to prove \( Q(n): \gcd(i_n, j_n) = \gcd(a, b) \) is true for all \( n \geq 0 \)

**Use Induction**

• **IB:** \( n = 0 \), \( Q(0): \gcd(i_0, j_0) = \gcd(a, b) \) is true as \( i \) and \( j \) are initialized to be \( a \) and \( b \) before entering the loop.

• **IH:** Assume \( Q(k): \gcd(i_k, j_k) = \gcd(a, b) \)

• **IS:** Show \( Q(k+1): \gcd(i_{k+1}, j_{k+1}) = \gcd(a, b) \)

By the assignment statements within the loop body, we know it computes \( i_k = q \cdot j_k + r_k \) \( \Rightarrow i_{k+1} = j_k \) and \( j_{k+1} = r_k \) (by property *)

then \( \gcd(i_{k+1}, j_{k+1}) = \gcd(i_k, r_k) = \gcd(i_k, j_k) \) (by IH)

and \( \gcd(i_k, j_k) = \gcd(a, b) \) (by IH)

Therefore, function GCD is correct !

---

**Modular Arithmetic**

\[ a \mod m = \frac{r}{y} \quad q = m \cdot \frac{a}{y} + \frac{r}{y} \]

• Let \( a \) be an integer and \( m \) be a **positive** integer. We denote the remainder \( r \), when \( a \) is divided by \( m \), by \( a \mod m \).

• Examples: \( q=2 \), \( \frac{a}{y} = 3 \), \( q = 0 \), \( \frac{a}{y} = -4 \)

\( 9 \mod 4 = 1 \), \( 9 \mod 3 = 0 \), \( 9 \mod 10 = 9 \), \( -13 \mod 4 = 3 \)

for \( x < 0 \), \( x \mod y = z+x \), where \( z \) is the smallest positive integer > \( |x| \) and is divisible by \( y \)

**Congruences**

• Let \( a \) and \( b \) be integers and \( m \) be a positive integer. We say that \( a \) is congruent to \( b \) modulo \( m \) if \( m \mid (a-b) \).

• We use the notation \( a \equiv b \pmod{m} \) to indicate that \( a \) is congruent to \( b \) modulo \( m \).

• We claim that \( a \equiv b \pmod{m} \) if and only if \( a \mod m = b \mod m \).
Examples:
Is it true that $46 \equiv 68 \pmod{11}$? Yes, because $11 \mid (46 - 68)$.
Is it true that $46 \equiv 68 \pmod{22}$? Yes, because $22 \mid (46 - 68)$.
For which integers $z$ is it true that $z \equiv 12 \pmod{10}$?
It is true for any $z \in \{\ldots, -28, -18, -8, 2, 12, 22, 32, \ldots\}$.

Theorem: Let $m$ be a positive integer. The integers $a$ and $b$ are congruent modulo $m$ if and only if there is an integer $k$ such that $a = b + km$.

Theorem: Let $m$ be a positive integer. If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then $a + c \equiv b + d \pmod{m}$ and $ac \equiv bd \pmod{m}$.

Proof:
We know that $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$ implies that there are integers $s$ and $t$ with $a = b + sm$ and $c = d + tm$.

Therefore, $(a + c) = (b + sm) + (d + tm) = (b + d) + (s + t)m$ and $ac = (b + sm)(d + tm) = bd + (bt + ds + stm)m$.
Hence, $a + c \equiv b + d \pmod{m}$ and $ac \equiv bd \pmod{m}$.

Theorem: Let $m$ be a positive integer. Then $a \equiv b \pmod{m}$ iff $a \mod m = b \mod m$.

Proof: Let $a = m*q_1 + r_1$, and $b = m*q_2 + r_2$ (A)

If part: show if $a \mod m = b \mod m$ then $a \equiv b \pmod{m}$

$a \mod m = b \mod m$ implies $r_1 = r_2$, therefore

$A_1 \rightarrow A_2 \rightarrow a - b = m(q_1 - q_2)$, thus $a \equiv b \pmod{m}$ by congruence def.

Only if part: show if $a \equiv b \pmod{m}$ then $a \mod m = b \mod m$

$a \equiv b \pmod{m}$ implies $a - b = m*q$, 
$m*q_1 + r_1 - (m*q_2 + r_2) = m*q$ by plugging (A) to the above
this yields $r_1 - r_2 = m(q - q_1 + q_2)$.

Since $0 \leq r_1, r_2 < m$, it must be that $0 \leq |r_1 - r_2| < m$.
The only multiple of $m$ in that range is $0$, i.e. $r_1 - r_2 = 0$.
Therefore $r_1 = r_2$, thus $a \mod m = b \mod m$. 
Introduction to Graphs

1. Definition: A **simple graph** $G = (V, E)$ consists of **vertex set** $V$, a nonempty set of **vertices/nodes**, and **edge set** $E$, a set of **edges** or **unordered pairs** of vertices. So each edge $e \in E$ is a set; $e = \{u, v\}$ where $u, v \in V$. An edge $e$ is a self-loop if $e = \{u, u\}$ for some $u \in V$.

2. Definition: A **undirected graph** is a simple graph with no self loops and there is **at most one edge** between two vertices (**multigraph** may have self-loops and multi-edges in between two vertices).

3. Definition: A **directed graph** $G = (V, E)$ consists of a vertex set $V$ and an edge set $E$. Edges are **ordered pairs or arrows** of elements in $V$. For each $e \in E$, $e = (u, v)$ where $u, v \in V$.

**In this chapter, we assume**

“no multi-edges” unless we clearly specify “with multi-edges”

**graphs mean undirected graphs**
Graph Models

• Example I: How can we represent a network of (bi-directional) railways connecting a set of cities?
  
• We should use an undirected graph with an edge \{a, b\} indicating a direct train connection between cities a and b.

\[ G=(V,E) \]
\[ V=\{B,T,N,W\} \]
\[ E=\{\{T,N\}, \{B,N\}, \{N,W\}\} \]

• Example II: In a round-robin tournament, each team plays against each other team exactly once. How can we represent the results of the tournament (which team wins over which other team)?
  
• We should use a directed graph with an edge \((a, b)\) indicating that team a wins over team b.

\[ G=(V,E) \]
\[ V=\{A,B,C,D\} \]
\[ E=\{(A,C),(B,C),(D,A), (D,B),(D,C)\} \]
Graph Terminology (undirected graphs)

- **Definition:** Two vertices $u$ and $v$ in an undirected graph $G$ are called **adjacent** or **neighbors** in $G$ if $\{u, v\}$ is an edge in $G$.

  If $e = \{u, v\}$, the edge $e$ is called **incident with** the vertices $u$ and $v$. The edge $e$ is also said to **connect** $u$ and $v$. The vertices $u$ and $v$ are called **endpoints** of the edge $\{u, v\}$.

- **Definition:** The **degree** of a vertex in an undirected graph is the number of edges incident with it, except that a loop at a vertex contributes twice to the degree of that vertex. In other words, you can determine the degree of a vertex in a displayed graph by **counting the lines** that touch it.

- A vertex of degree 0 is called **isolated**, since it is not adjacent to any vertex.

### The Handshaking Theorem

- **Theorem:** Let $G = (V, E)$ be an undirected graph with $|E|$ edges. Then $2|E| = \sum_{v \in V} \deg(v)$ // sum of degrees

- **Example:** How many edges are there in a graph with 10 vertices, each of degree 6?

  Solution: The sum of the degrees of the vertices is $6 \cdot 10 = 60$. According to the Handshaking Theorem, it follows that $2e = 60$, so there are 30 edges.

- **Theorem:** An undirected graph has an even number of vertices of odd degree.

  **Proof:** Let $V1$ and $V2$ be the set of vertices of even and odd degrees, respectively. Then by Handshaking theorem

  $$2|E| = \sum_{v \in V} \deg(v) = \sum_{v \in V1} \deg(v) + \sum_{v \in V2} \deg(v)$$

  Since both $2|E|$ and $\sum_{v \in V1} \deg(v)$ are even, $\sum_{v \in V2} \deg(v)$ must be even $\Rightarrow |V2|$ is even
And for directed graphs:

- **Definition**: When \((u, v)\) is an edge of the directed graph \(G\), \(u\) is said to be **adjacent to** \(v\), and \(v\) is said to be **adjacent from** \(u\).

- The vertex \(u\) is called the **initial vertex** of \((u, v)\), and \(v\) is called the **terminal vertex** of \((u, v)\). The initial vertex and terminal vertex of a loop are the same.

- **Definition**: In a directed graph, the **in-degree** of a vertex \(v\) is the number of edges with \(v\) as their terminal vertex. The **out-degree** of \(v\) is the number of edges with \(v\) as their initial vertex.

- **Question**: How does adding a loop to a vertex change the in-degree and out-degree of that vertex?
  
  **Answer**: It increases both the in-degree and the out-degree by one.

---

**Example**: What are the in-degrees and out-degrees of the vertices \(a, b, c, d\) in this graph:

\[
\begin{align*}
\deg^\text{in}(a) &= 1 \\
\deg^\text{out}(a) &= 2 \\
\deg^\text{in}(b) &= 4 \\
\deg^\text{out}(b) &= 2 \\
\deg^\text{in}(d) &= 2 \\
\deg^\text{out}(d) &= 1 \\
\deg^\text{in}(c) &= 0 \\
\deg^\text{out}(c) &= 2
\end{align*}
\]

**Theorem**: Let \(G = (V, E)\) be a directed graph. Then: \(\sum_{v \in V} \deg^\text{in}(v) = \sum_{v \in V} \deg^\text{out}(v) = |E|\)

- This is easy to see, because every new edge increases both the sum of in-degrees and the sum of out-degrees by one.
Special Graphs

- **Definition:** The complete graph on \( n \) vertices, denoted by \( K_n \), is the simple graph that contains exactly one edge between every pair of distinct vertices.

- **Definition:** The \( n \)-cube, denoted by \( Q_n \), is the graph that has vertices representing the \( 2^n \) bit strings of length \( n \). Two vertices are adjacent if and only if the bit strings that they represent differ in exactly one bit position.

- **Definition:** A graph is called bipartite if its vertex set \( V \) can be partitioned into two disjoint nonempty sets \( V_1 \) and \( V_2 \) such that every edge in the graph connects a vertex in \( V_1 \) with a vertex in \( V_2 \) (so that no edge in \( G \) connects either two vertices in \( V_1 \) or two vertices in \( V_2 \)).
• For example, consider a graph that represents mating partners of penguin in a colony. (March of Penguins!)

• This graph is bipartite, because the vertex set can be partitioned into female set and male set then each edge connects a vertex between the two sets.

• **Example I:** Is graph G below bipartite?

  No, because there is no way to partition the vertices into two sets so that there are no edges with both endpoints in the same set.

• **Example II:** Is graph G below bipartite?

  Yes, because we can display G like this:

• **Definition:** The **complete bipartite graph** $K_{m,n}$ is the graph that has its vertex set partitioned into two subsets of $m$ and $n$ vertices, respectively. Two vertices are connected if and only if they are in different subsets.

**Neural Network!**
Operations on Graphs

- **Definition:** A subgraph of a graph $G = (V, E)$ is a graph $H = (W, F)$ where $W \subseteq V$ and $F \subseteq E$.
- **Note:** $H$ must be a valid graph. i.e. For each $\{u, v\}$ in $F$, $u$ and $v$ are vertices in $W$.
- **Example:**

```
K_5
```

```
subgraph of K_5
```

- **Definition:** The union of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the simple graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2$. 

```
(\{v_1, e_1\}, \{v_2, e_2\})
```

- The union of $G_1$ and $G_2$ is denoted by $G_1 \cup G_2$.

```
G_1
```

```
G_2
```

```
G_1 \cup G_2 = K_5
```
Representing Graphs

- **Definition:** Let $G = (V, E)$ be a graph with $|V| = n$. Suppose that the vertices of $G$ are listed in arbitrary order as $v_1, v_2, \ldots, v_n$.

- The **adjacency matrix** $A_G$ of $G$, with respect to this listing of the vertices, is the $n \times n$ zero-one-valued matrix with 1 as its $(i, j)$-th entry when $v_i$ and $v_j$ are adjacent, and 0 otherwise.

- In other words, for an adjacency matrix $A = [a_{ij}]$, $a_{ij} = 1$ if $\{v_i, v_j\}$ is an edge of $G$, $a_{ij} = 0$ otherwise.

- **Example:** What is the adjacency matrix $A_G$ for the following graph $G$ based on the order of vertices $a, b, c, d$?

Solution: 

$$
A_G = \begin{bmatrix}
0 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 \\
\end{bmatrix}
$$

Definition: **Adjacency list** representation consists of a list of its vertices together with a separate list for each vertex that contains all the vertices adjacent to that vertex.

Example: let $1 = a$, $2 = b$, $3 = c$ and $4 = d$

<table>
<thead>
<tr>
<th>Vertex</th>
<th>List of adjacencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2, 3, 4</td>
</tr>
<tr>
<td>2</td>
<td>1, 4</td>
</tr>
<tr>
<td>3</td>
<td>1, 4</td>
</tr>
<tr>
<td>4</td>
<td>1, 2, 3</td>
</tr>
</tbody>
</table>

Note: Adjacency Matrix and Adjacency Lists can be used to defined directed graphs also.