8.2 Directed Graphs (Digraphs)

Finite Math Graph Memories:

Chapter 5: Graph Model

Chapter 6: Complete Graph \( K_n \)
Each of \( N \) vertices is connected to every other vertex in the graph.

Chapter 7: Trees
A connected graph with no circuits.

Chapter 8: DIGRAPHS ("DIRECTIONAL GRAPH")
A graph with direction associated with the edges.

**Euler Path/Circuit:**
Covers each edge exactly once. Vertices may be visited multiple times.

**Hamilton Path/Circuit:**
Covers each vertex exactly once. Many edges are left untraveled.

**Minimum Spanning Tree:** The spanning subgraph of a network with the least weight.
**Steiner Tree:** The shortest possible network, if it exists. (Steiner tree forms three 120 degree angles at an interior junction point.)

**Similarities?**
- vertices
- could still travel (ch5/b)
- circuits

**Differences?**
- edges are arrows
- have to travel in direction of arrow
DIGRAPh
A graph in which the edges have a DIRECTION associated with them is called a directed graph or “digraph” for short. Digraphs are used to represent precedence relationships in scheduling problems.

ARC
An edge with direction associated with it. Order matters!

\[
\text{Arc } XY \quad X \rightarrow Y \\
\text{Arc } YX \quad X \leftarrow Y
\]

Example. Draw the digraph with vertex set \( V = \{G, H, O, S, T\} \) and arc set \( A = \{GH, GO, OH, ST, TG, TH\} \).

PATH
A sequence of arcs that can only be traveled in the direction indicated and no arc appears more than once. Like Che5

Example:

\[\text{E A B D (lots of possible)}\]

“circuit
CYCLE
A path that starts and ends at the same vertex is a cycle in the digraph.

Example:

\[\text{C E A C (lots of possible)}\]

INCIDENT TO/FROM
If arc AB is an arc in a digraph, then we say:

vertex A is incident to vertex B (think: "pointing to")

vertex B is incident from vertex A (think: "coming from")

ADJACENT ARCS
Arc #1 is adjacent to Arc #2 if one Arc #1 BEGINS where Arc #2 ENDS.

*** IN OTHER WORDS... An arc is adjacent to your arc if it's LAST letter is your arcs FIRST letter. What would come right before it?****

Example:

CD is adjacent to AC.

CD is NOT adjacent to CE.
Example: Consider the digraph:

(a) find all vertices incident to B  A, D
(b) find all vertices incident from B  E, C
(c) find an arc adjacent to AB  BC, BE
(d) find an arc adjacent to ED  DB, BE, AE
(e) find any path from A to D  AED, AD, ABCD, ABD
(f) find any cycle in the digraph  DBED, BCDB

OUTDEGREE/INDEGREE

out-degree of vertex A is the number of arcs that have A as a start
in-degree of vertex A is the number of arcs that have A as an end

Example: outdegree(C) = 3
outdegree(D) = 0
indegree(C) = 1
indegree(D) = 3

** VOCAB ACTIVITY**

How well do you know these new words?

Applications of Digraphs

Traffic Flow: Graph Models of cities where vertices represent intersections and arcs represent one-way streets. A two way street would have two arcs, one for each direction.

Telephone Traffic: Telephone companies use “call digraphs” to analyze calls to and from numbers in their network.

Tournaments: Teams are vertices. Outcomes of games are arcs. AB “A defeated B”

Organization Charts: “Who’s the Big Cheese?” Vertices are people, arcs represent who is the immediate boss – a “chain of command” model.

Scheduling: Organizes the precedence relationships in a scheduling problem so we have a visual reminder of what tasks must be completed first. We call these “project digraphs”.

Scanned by CamScanner
Finite Math A, Chapter 8: Scheduling

Example: Suppose the vertices in the following digraph represent basketball teams and that an arc from \( X \) to \( Y \) means "\( X \) defeated \( Y \)" in a game.

Which team(s) won the most games?

\[ C \text{ and } D \text{ both won 3} \]

Which team(s) won the fewest games?

\[ B \]

Did any teams play each other more than once?

\[ A \neq B \neq D \neq E \]

Example: The course catalog at your college gives the following information regarding math classes.

A prerequisite for Calculus III is Calculus II. A prerequisite for Calculus II is Calculus I. Linear Algebra can only be taken after both Calculus II and Fundamentals of Mathematical Proofs have been taken. A prerequisite for Fundamentals of Mathematical Proofs is Calculus I. In order to take Real Analysis I, a student must have completed Calculus III and Linear Algebra. A prerequisite for Real Analysis II is Real Analysis I.

Draw a digraph which models the relationships between the math classes.

House Building Example: See pages 280 & 281
3.1 Basic Elements of Scheduling:

**PROCESSOR:**  WORKER  
Whomever or whatever is working on a task  
- Use P_1, P_2, ..., P_n, etc.

**TASK:**  WORK  
A task is an indivisible unit of work that cannot be broken up into smaller units (always carried out by a single processor.)  
- Use Capital Letters: A, B, C, etc.. or maybe PL (Plumbing) or EP (Exterior Painting)

A task that cannot be started yet:  INELIGIBLE  
A task that is presently being carried out:  IN EXECUTION  
A task that can be started now:  READY  
A task that is done:  COMPLETED

**PROCESSING TIME:**  The amount of time it takes a task to be completed by one processor  
Assume “Robotic Behavior” or humans trained to be very standardized

Notation:  X(y) = It takes y amount of time for one processor to complete task X  
Example:  W(15) = It takes 15 minutes to Wash the windows

**PRECEDENCE RELATIONS:**  
A situation where one or more tasks must be FINISHED before another task BEGINS.

In our house example: what is an example of a precedence relationship?  
build walls → paint walls

**INDEPENDENT TASKS**  
Tasks with no precedence relations between them.

Example:  wash dishes + mow lawn

Precedence Relation:

\[
\begin{align*}
X & \rightarrow Y \\
X & \quad | \\
Y & \quad | \\
\rightarrow & \quad | \\
O & \quad | \\
\end{align*}
\]

X\&Y Independent

\[
\begin{align*}
X & \rightarrow Y \\
X & \quad | \\
Y & \quad | \\
\rightarrow & \quad | \\
O & \quad | \\
\end{align*}
\]

X → Y and Y → Z

\[
\begin{align*}
X & \rightarrow Y \\
X & \quad | \\
Y & \quad | \\
\rightarrow & \quad | \\
Z & \quad | \\
\end{align*}
\]

implies X → Z
In this chapter, we will use PROJECT DIGRAPHS to help us OPTIMIZE SCHEDULING PROBLEMS.

Project Digraph:
Example: Look at the following PROJECT DIGRAPH

What are some precedence relationships?
- dry hair before fix hair
- shower before dress

Name a pair of independent tasks
- eat breakfast & dry hair

Note: A digraph has no start and end. A project digraph has a start(0) and end (0).
Start and End are "imaginary tasks" that just help you "see" the path through the tasks. -- Think of Start as "cutting the ribbon" and End as declaring the project complete.

Scheduling Problem
Example:
You wreck your car. There are 4 repairs. There are two mechanics. Exterior body work must be completed before painting and exterior work can begin (A→C)
(A) exterior body work: 4 hrs;  (B) engine repairs: 5 hrs;
(C) painting and exterior work: 7 hrs;
(D) transmission repair: 3 hrs

Possible Schedules:

Seems like we could do better:

<table>
<thead>
<tr>
<th>Time: 0 1 2 3 4 5 6 7 8 9 10 11 12 13</th>
</tr>
</thead>
<tbody>
<tr>
<td>P₁ (A(4)  D(3)  Idle)</td>
</tr>
<tr>
<td>P₂ (B(5)  C(7)  Idle)</td>
</tr>
</tbody>
</table>

Fin=12 hrs

A little better...

<table>
<thead>
<tr>
<th>Time: 0 1 2 3 4 5 6 7 8 9 10 11 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>P₁ (A(4)  B(5)  Idle)</td>
</tr>
<tr>
<td>P₂ (D(3)  C(7)  Idle)</td>
</tr>
</tbody>
</table>

Fin=11 hrs

Different, same Fin = 11 time.

<table>
<thead>
<tr>
<th>Time: 0 1 2 3 4 5 6 7 8 9 10 11 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>P₁ (A(4)  B(5)  C(7)  Idle)</td>
</tr>
<tr>
<td>P₂ (D(3)  Idle)</td>
</tr>
</tbody>
</table>

Due to the precedence relation A → C. WE CANNOT DO ANY BETTER THAN Fin = 11 hrs.
This mean 11hrs is our CRITICAL TIME FOR THIS PROJECT. Every project has an absolute minimum time.

Our goal is to find this

"OPTIMIZE"
Example: Building a Dream Home on Mars
You have earned a trip to Mars and will be living in a MHU (Martian Habitation Unit).
The MHU must be assembled by special construction robots that can be rented by the hour. The precedence relationships in the chart must be followed. You have to tell the robots when to do what. How many should you rent? What is the quickest way to get the job done?

<table>
<thead>
<tr>
<th>Task</th>
<th>Label (P-time)</th>
<th>Precedent tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assemble pad</td>
<td>AP(7)</td>
<td>READY!</td>
</tr>
<tr>
<td>Assemble flooring</td>
<td>AF(5)</td>
<td>READY!</td>
</tr>
<tr>
<td>Assemble wall units</td>
<td>AW(6)</td>
<td>READY!</td>
</tr>
<tr>
<td>Assemble dome frame</td>
<td>AD(8)</td>
<td>READY!</td>
</tr>
<tr>
<td>Install floors</td>
<td>IF(5)</td>
<td>AP, AF</td>
</tr>
<tr>
<td>Install interior walls</td>
<td>IW(7)</td>
<td>IF, AW</td>
</tr>
<tr>
<td>Install dome frame</td>
<td>ID(5)</td>
<td>AD, IW</td>
</tr>
<tr>
<td>Install plumbing</td>
<td>PL(4)</td>
<td>IF</td>
</tr>
<tr>
<td>Install atomic power plant</td>
<td>IP(4)</td>
<td>IW</td>
</tr>
<tr>
<td>Install pressurization unit</td>
<td>PU(3)</td>
<td>IP, ID</td>
</tr>
<tr>
<td>Install heating units</td>
<td>HU(4)</td>
<td>IP</td>
</tr>
<tr>
<td>Install commode</td>
<td>IC(1)</td>
<td>PL, HU</td>
</tr>
<tr>
<td>Complete interior finish work</td>
<td>FW(6)</td>
<td>IC</td>
</tr>
<tr>
<td>Pressurize dome</td>
<td>PD(3)</td>
<td>HU</td>
</tr>
<tr>
<td>Install entertainment unit</td>
<td>EU(2)</td>
<td>PU, HU</td>
</tr>
</tbody>
</table>

If we make a project digraph it is much easier to see which tasks are “ready” from the start and which tasks we will have to wait to start.

Which tasks could be started right away?

AP, AF, AW, AD

Example:
Draw a project digraph for a project consisting of the six tasks described by the following table:

<table>
<thead>
<tr>
<th>Task</th>
<th>Processing Time</th>
<th>Precedence Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
<td>Ready!</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
<td>Ready!</td>
</tr>
<tr>
<td>C</td>
<td>5</td>
<td>A, B</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>C</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
<td>D</td>
</tr>
<tr>
<td>F</td>
<td>8</td>
<td>B</td>
</tr>
</tbody>
</table>

FROM TABLE

A → C✓
B → C✓
C → D✓
D → E✓
B → F✓

Start (0)

Must Do These First!