

A group of people are gathered around a wooden table, looking at a large sheet of architectural blueprints. The scene is dimly lit, with warm, golden light coming from the side, creating a focused and collaborative atmosphere. Several hands are visible, pointing at different parts of the blueprint. The people are wearing casual to business-casual attire, including a plaid shirt, a beige sweater, and a dark jacket. The background is blurred, showing what appears to be an office or meeting room setting.

Lecture 5: Planning

CS 5150, Spring 2026

Administrative reminders

- Schedule a meeting with your client
 - See Canvas homepage for course staff details
- Project plan (today's topic) due ~~Feb 5~~ **Feb 10 (Tue)**
 - In-class activity: ~~Feb 5~~ **Feb 10**
- Some teams are still looking for members (See Ed)

Project evaluation

Project evaluation/Grading Items

- Reports: Pitch, Plan, Sprint Report, Final report
- Presentations: Midpoint, Final
- Client meeting (at least once every sprint)
 - Your client will grade your interactions
- Peer Review (once per sprint)
- Code Quality/Software Engineering Practices (CI, Reviews, etc.)
 - Tool reports, manual reviews

Peer review

Rubric will evaluate the following:

- **Professionalism**

- Attendance/promptness
- Preparation for meetings
- Respectful interactions

- **Quality of contributions**

- **Effective communication**

- Reports problems early
- Communicates clearly

- **Initiative**

- Contributes to brainstorming
- Volunteers for tasks

- **Healthy team dynamics**

- Raises and resolves conflicts
- Trusts other members, helps them grow

Feasibility

Should you or your client commit to a project?

- Is it possible?
 - Could you do it? (resource availability)
 - How long would it take?
 - How much would it cost?
 - Who benefits?
 - How risky is it?
 - What are the alternatives?
- *Before* committing, conduct a study to inform a **go/no-go** decision
 - Result may be a **proposal** or lead to a **budget request**

Feasibility studies are difficult

- Must grapple with **uncertainty**
 - Unclear scope
 - Uncertain or hard-to-quantify benefits
 - Rough estimates of resource requirements and timetable
 - Technical approach may not pan out
 - Organizational changes may be required
 - Opportunity costs
- Rely on judgement of experienced people
- Early mistakes are the most costly
- Need to **advocate** to build support
 - But beware risk distortion, ulterior motives

Example: US government agency

Objective: Transition from filing paper records to managing digital documents

Chronology:

1. Request **prototype** from university to demonstrate core technology
2. Congress **approves funds** to procure major computer system
3. Commission external **feasibility study** on technical approach, results of prototype

Do we see a problem?

Organizational problems [Example]

- Agency senior management lacked experience to lead such a large, transformational project
- Agency had not considered workflow and job changes that would affect all employees
- Agency had not quantified data volume or specified access policies
 - Stewards of private & classified information

⇒ Requirements are complex and only partially understood

Which methodology would you recommend?

Pollev.com/cs5150sp26

Next steps [Example]

- External study was officially limited to technical aspects, but:
 - Noted that success required major changes in senior management
 - Recommended phased approach with iterative refinement (over several years), after addressing organizational problems
- Agency did not want to return money to Congress
 - Adopted pure waterfall model
 - Issued RFP based on current understanding of requirements
 - Signed major contract with contractor
 - System encountered numerous problems during prolonged implementation

What must be considered?

- Scope
- Approach
- Methodology
- Resources
- Schedule
- Risks
- Alternatives

Should be documented in your
Project Plan

See:

<https://www.cs.cornell.edu/courses/cs5150/2026sp/projects/project-plan.html>

Scope clarification

- Define the boundaries of the system
 - List of included features
 - List of excluded features
 - List of dependencies
 - List of current systems being replaced
- Confusion over scope often leads to **client dissatisfaction**
 - "Is that all you planned to do?"
 - "But your product isn't useful without XYZ."
- Should also review existing systems (including competitors')

Technical approach

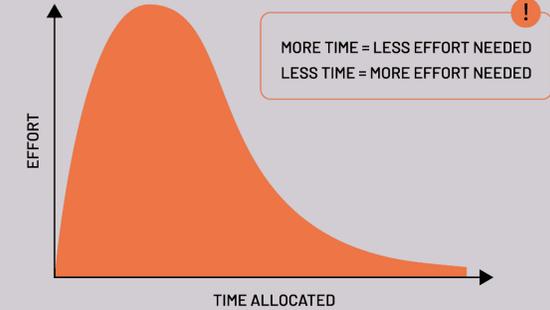
- Proposed system must be technically feasible
 - **Estimates** of **scale** (number of users, volume of data, rate of transactions)
 - Identify features that require **research** (no standard solution) or **new team expertise** (domain-specific knowledge, advanced techniques)
- Analyze a viable design to estimate resources, schedule
 - Tentative **system architecture** (data storage, UI context, deployment infra)
- Approach used by final product may be very different

Estimation

Essential skill for efficient planning

- Expected load
- Anticipated infrastructure needs
- Performance bottlenecks in candidate architectures
- Duration of tasks
- Cost/benefit tradeoffs

Parkinson's Law



LifeHack

Estimation for scheduling

- Estimating task duration is *very* difficult, but can be improved with **feedback**
 - Estimate effort of task before starting
 - Keep a log of time spent on each task (design, documentation, implementation, testing, review)
 - Compare log to estimate when closing out tasks
 - Keep a log of time spent on other aspects of project (meetings, training, reports, reviews)
- Do not evaluate effort until task is 100% complete
 - Large gap between "not done" and "done"
 - Documentation
 - Tests
 - Cleanup
 - Review
 - Beware **Parkinson's Law**: "Work expands to fill the time available"
- Don't neglect startup time

Project management

The operating system for the IBM 360 was **two years late**.

How does a project get two years behind schedule?

One day at a time. - Fred Brooks, Jr. *The Mythical Man-Month*

Milestones & deliverables

Deliverable

- Work product provided to the client
 - Mock-up
 - Demonstration
 - Prototype
 - Report
 - Presentation
 - Documentation
 - Code
- Release of a system or subsystem to customers and users

Milestone

- Completion of a predetermined set of activities
 - Delivery of a deliverable
 - Completion of a process step
 - End of a sprint
 - Reaching a testing target
- An internal goalpost, useful for monitoring progress against the schedule
 - Provides visibility

Outline plan

- Your first report requires an outline plan
 - Preliminary timetable
 - Process steps (or sprints)
 - Milestones (including deliverables)
 - Decision points
 - Interactions with and dependencies on external parties
- Plan should be reviewed, refined, revised regularly

Scheduling activities

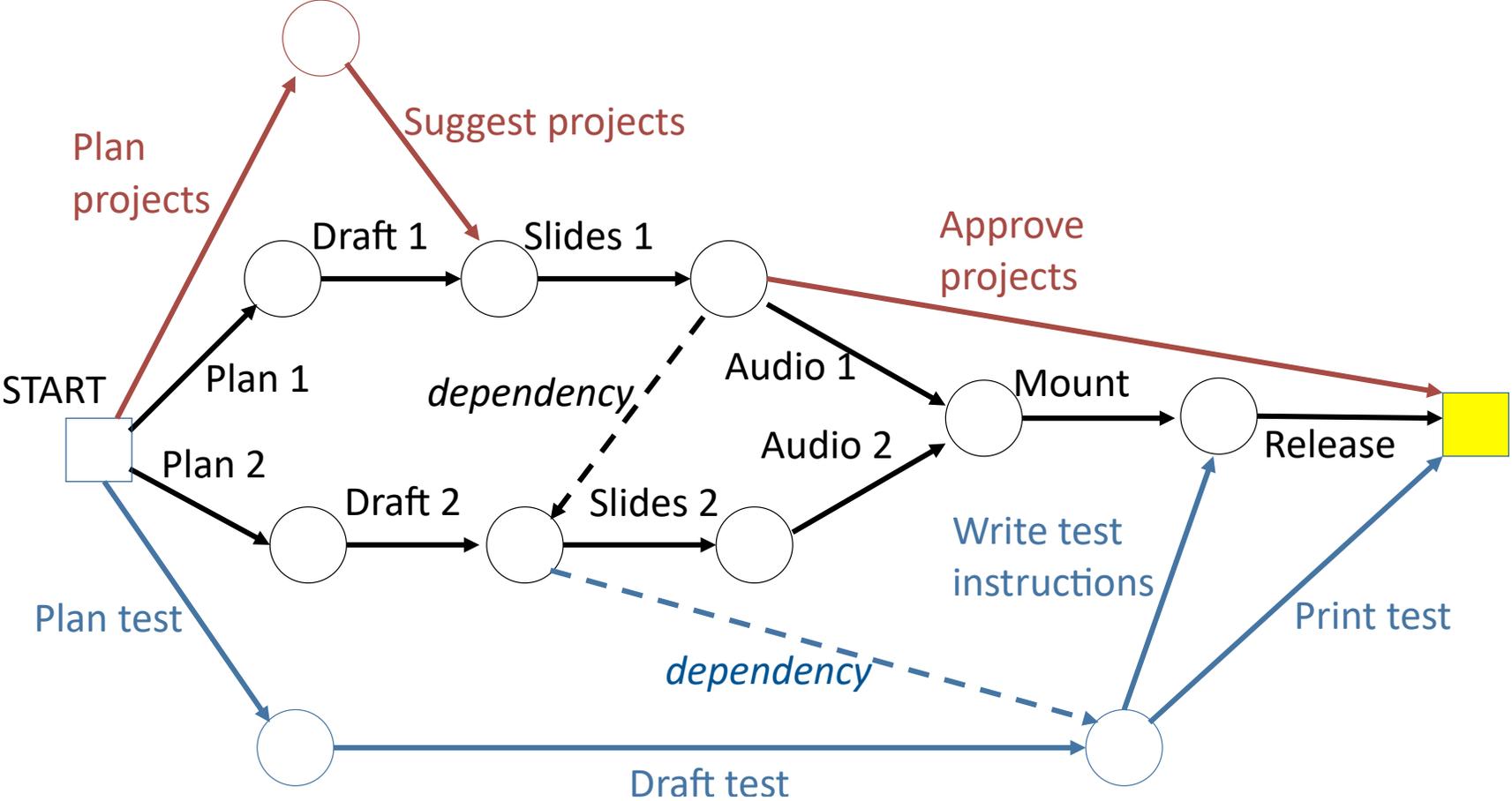
- Inputs:
 - Duration
 - Dependencies
 - Resources

Task	Dependencies	Duration
A		3 days
B	A	4 days
C	B	2 days
D	A	1 day
E		5 days
F	A, E	3 days
G	F	3 days
(Finish)	C, G	

Activity networks

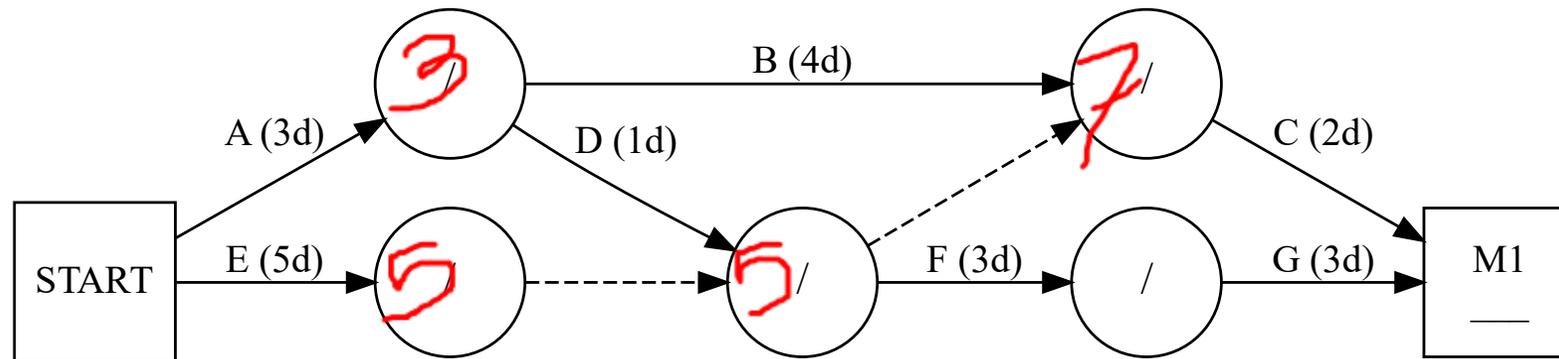
- Dependencies are an adjacency list – defines a graph
 - Large projects often have >10,000 activities
- Nodes are events (start and end of activities)
- Activities are weighted edges
- Dependencies are zero-weight edges

Example: Distance learning course milestone



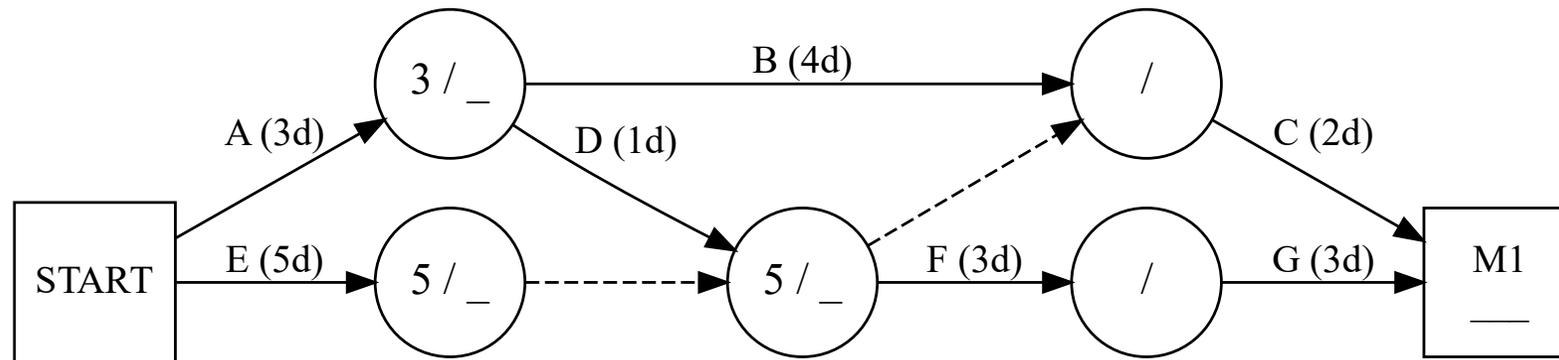
Critical path analysis I

- Earliest event dates
 - For each inbound edge, add edge's weight to activity's earliest start date to get activity's earliest completion date
 - Outbound edges' earliest start date is maximum of inbound edges' earliest completion dates
- Yields earliest project completion date



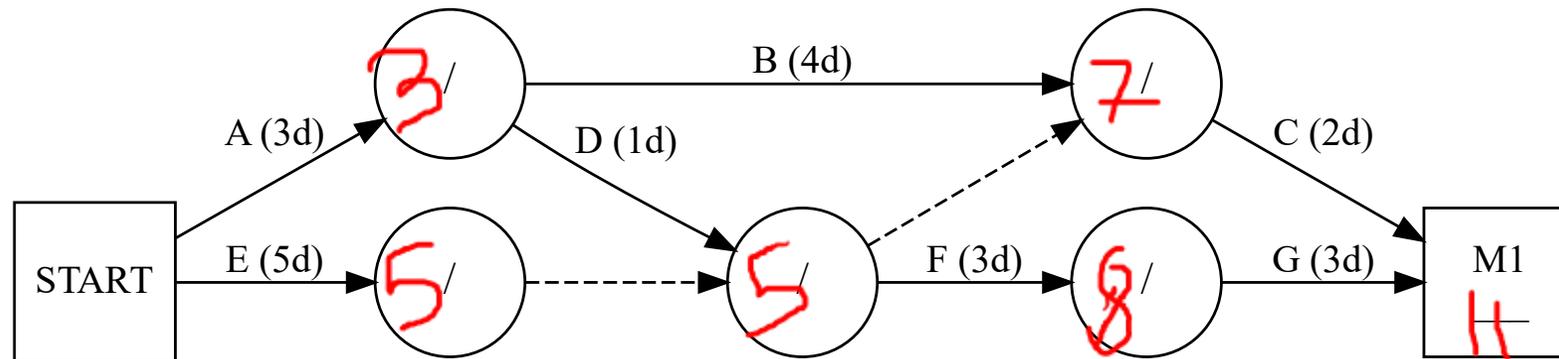
What is M1's earliest completion date?

PollEv.com/cs5150sp26



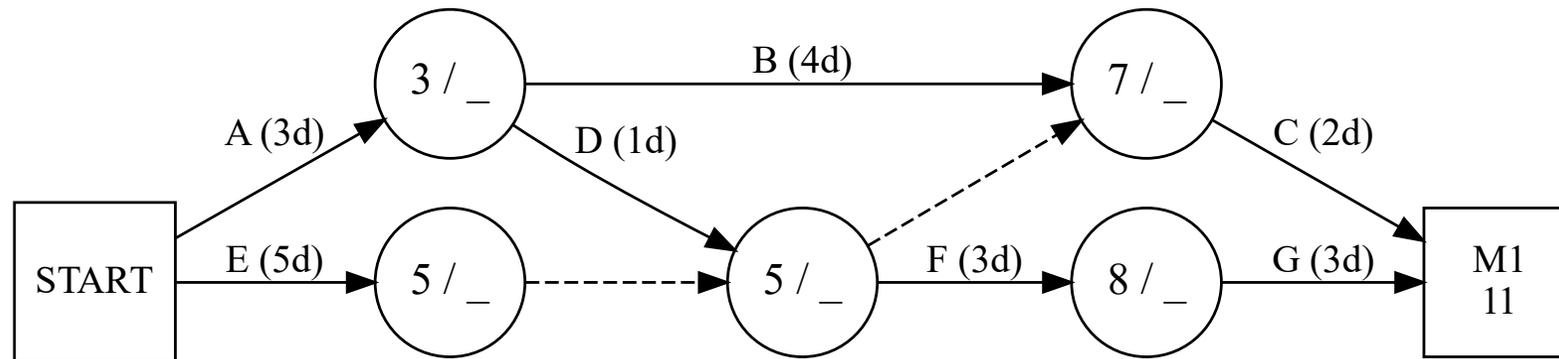
Critical path analysis I

- Earliest event dates
 - For each inbound edge, add edge's weight to activity's earliest start date to get activity's earliest completion date
 - Outbound edges' earliest start date is maximum of inbound edges' earliest completion dates
- Yields earliest project completion date



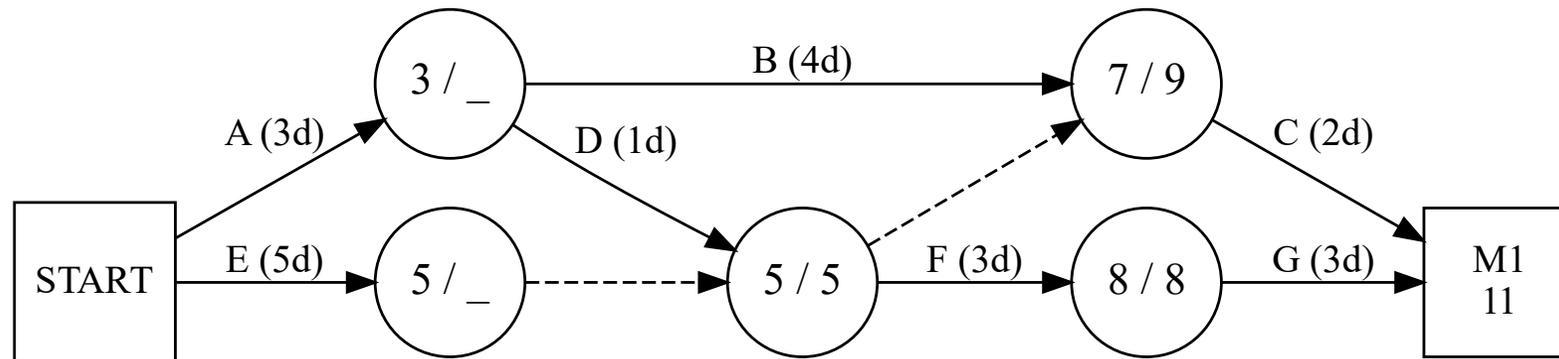
Critical path analysis II

- Latest event dates
 - How late can activities start while still achieving earliest project completion date?
 - Given a node with a latest event date, subtract weight for each incoming edge to get activities' latest start dates
 - Latest event date is minimum of outgoing edges' latest start dates



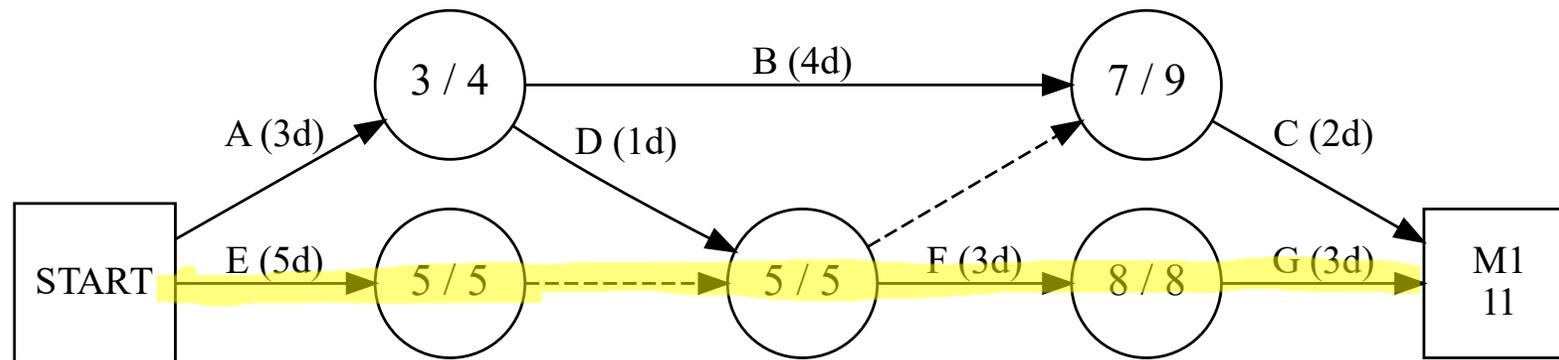
What is the slack in finishing activity A?

PollEv.com/cs5150sp26



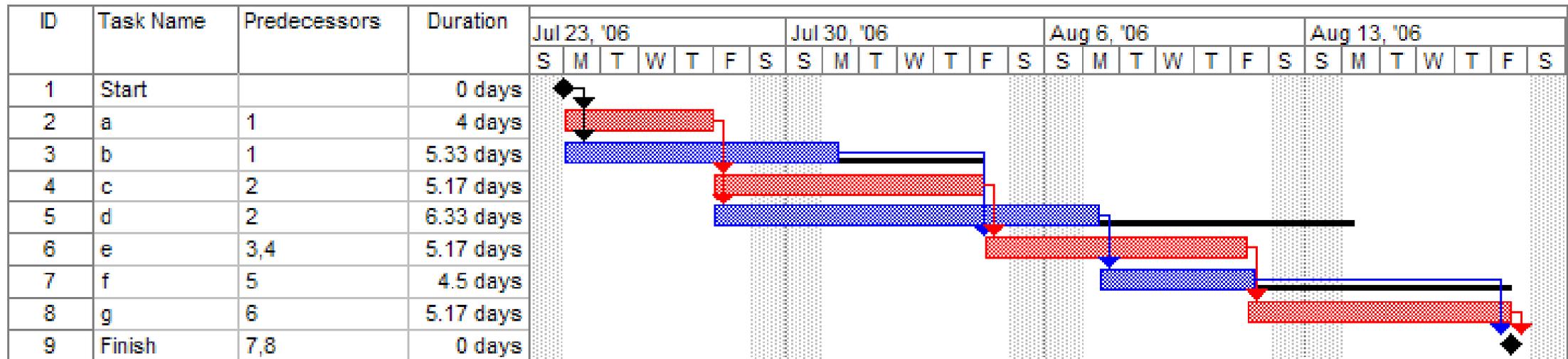
Critical path analysis III

- **Slack**: difference between an activity's earliest start date and latest end date, minus duration
- **Critical path**: all events along path have earliest event date == latest event date
 - All activities along critical path have zero slack

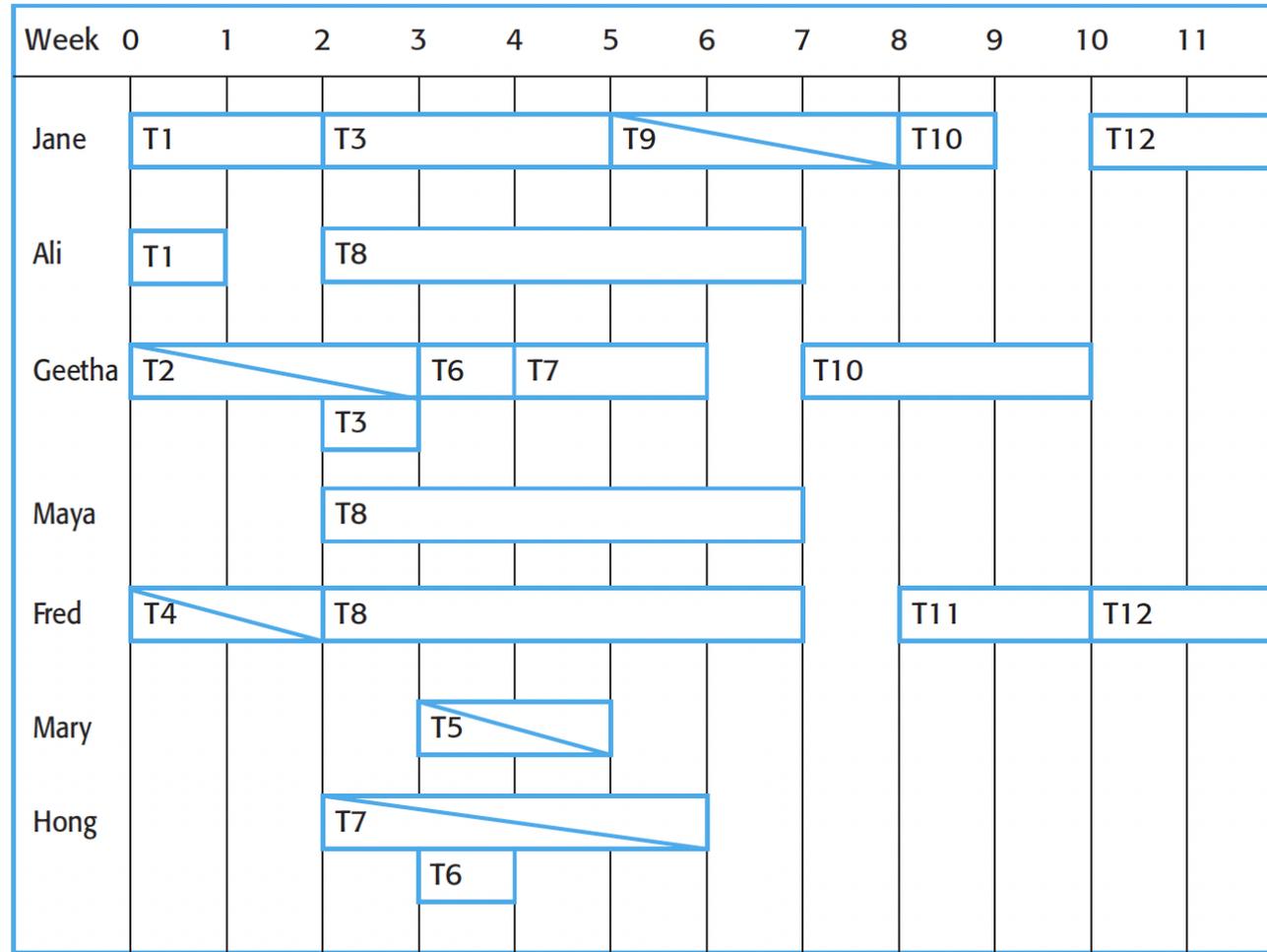


Gantt charts

- Visualize plan for when activities actually take place
 - Number of parallel activities limited by resources
- Can highlight critical path, slack time



Staff Allocation Chart



Risk management

1. Identify risks
 - Brainstorm what could go wrong
2. Analyze risks
 - Determine likelihood and consequence
 - Prioritize based on "risk"
3. Plan
 - Avoidance: reduce likelihood
 - Mitigation: reduce consequence
 - Contingency: "Plan B"
4. Monitor
 - Update risks regularly

RISK		CONSEQUENCE			
		CATASTROPHIC	CRITICAL	MARGINAL	NEGLIGIBLE
PROBABILITY	FREQUENT	<u>Unacceptable</u>	<u>Unacceptable</u>	<u>Unacceptable</u>	<i>Undesirable</i>
	PROBABLE	<u>Unacceptable</u>	<u>Unacceptable</u>	<i>Undesirable</i>	Tolerable
	OCCASIONAL	<u>Unacceptable</u>	<i>Undesirable</i>	Tolerable	Tolerable
	REMOTE	<i>Undesirable</i>	Tolerable	Tolerable	Negligible
	IMPROBABLE	Tolerable	Tolerable	Negligible	Negligible
	INCREDIBLE	Negligible	Negligible	Negligible	Negligible

CS 5150 project plans

- Feasibility studies and project plans should be **written**
 - Well-written, well-presented – review entire document
 - Short enough that everybody reads it
 - Long enough that no important topics are skipped
 - A report that is not read and understood is not useful
- Keep in mind:
 - Team availability, team skills
 - Time constraints
 - Equipment and software
 - Start-up time
 - Client availability
 - Scope (not vague, not too ambitious)