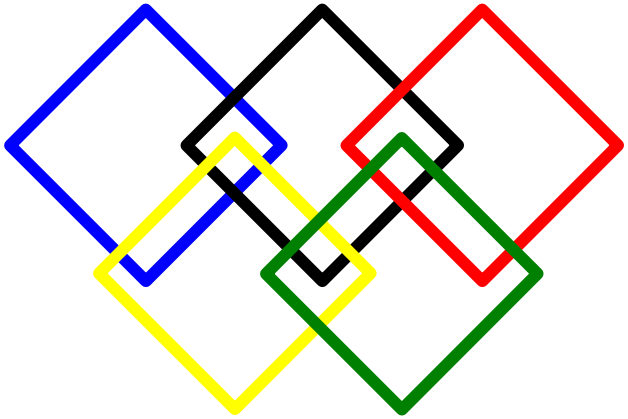


CSCI 1108

Project Management

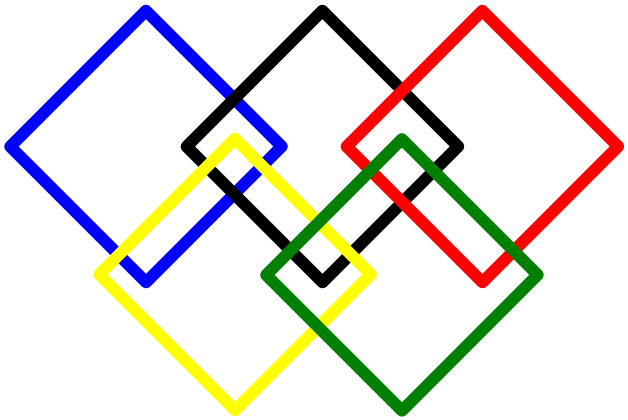
Today's Topics

- The Project:
  - Qualifying Event for Robot Olympics
- Program Planning
  - Strategy
  - Tactics
- Project Management [1]
  - Motivation
  - What's to manage
  - Scheduling tasks
  - Gantt charts
  - Managing risk

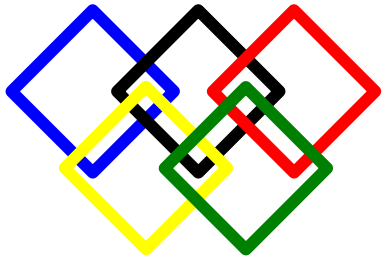
[1] https://ocw.mit.edu/courses/mechanical-engineering/2-000-how-and-why-machines-work-spring-2002/tools/management.pdf

Qualifying Event for Robot Olympics

- The event:
  - Line Hurdles



- Your team's Tasks:
  - Write a program for the event
  - Compete in the Qualifying Event
  - Write a report on your project

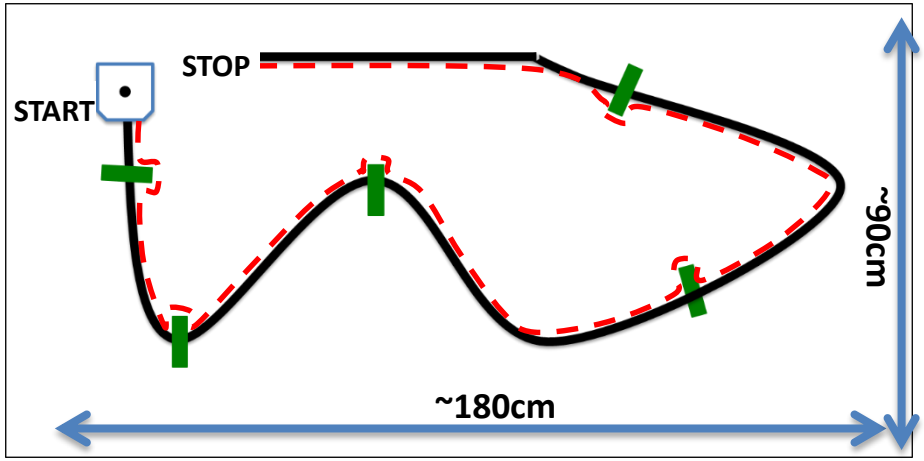


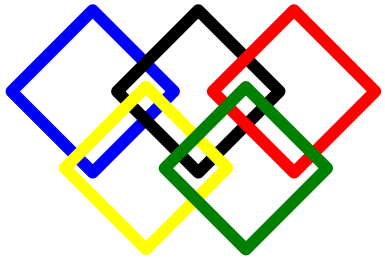
General Rules

- One program for the event
  - Program cannot be changed once competition begins
- No human interference
  - You may not touch a robot while it is competing
  - Robots may be disqualified if interference occurs
- Robot's performance affects your grades
  - See project specifications for rubric

Line Hurdles

- As quickly as possible
  - Race to the end of the line
  - Go around the green blocks
  - Do not dislodge the blocks
- Robot has three 1-minute attempts





The Project Report

General Information

- Report is aimed at peers, TAs, & instructor
- 5 pages, 11pt (see template)
- The report must
  - Provide sufficient background
  - Describe the program design, strategy, and tactics
  - **Justify your design decisions**
  - Describe how successful the programs were
  - State overall conclusions
- Rubric in project specification

Recommended Structure

- Title and author information
- Abstract
- Introduction
- Background
- Main Body
  - Line Hurdles
- Results
- Conclusions and Future Work
- References

Deliverables

- One Program
  - Loaded on your robot to compete in the Qualifying event for the Robot Olympics.
  - Must be submitted via **csci1108@gmail.com** before your presentation period (**October 11/12**)
- Technical Report in PDF or Word format
  - 5 pages
  - In hard copy (start of the next Lab following the presentation period)
  - Must be submitted via **csci1108@gmail.com** (**October 13/14**)

Where Do We Start???

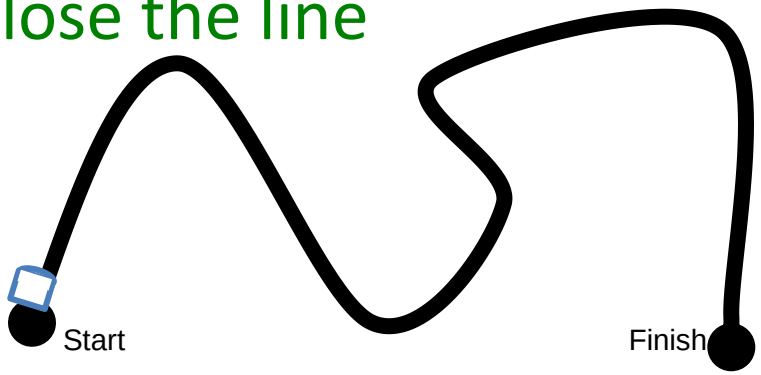
- Situation:
  - 2 Labs (+ overtime if need be)
  - 1 Program
  - 1 Project Report
  - 3 to 5 group members
  - 1 Robot
- Step 1: Identify the Tasks
  - **Develop a program**
  - Write a project report

Steps for Developing a Program

1. Develop program *strategy*
2. Identify *tactics* to implement the strategy
3. **Model tactics with state transition diagrams**
4. Implement program based on STDs
5. Test your program
6. Refine strategy and tactics as necessary
7. Repeat

Strategy

- How are we going to solve the problem?
  - Typically there is more than one way
  - Can be described in a couple sentences
- Example: The Line Race
  - Go as quickly as possible, and pay the price of losing the line
  - Go slow enough and never lose the line

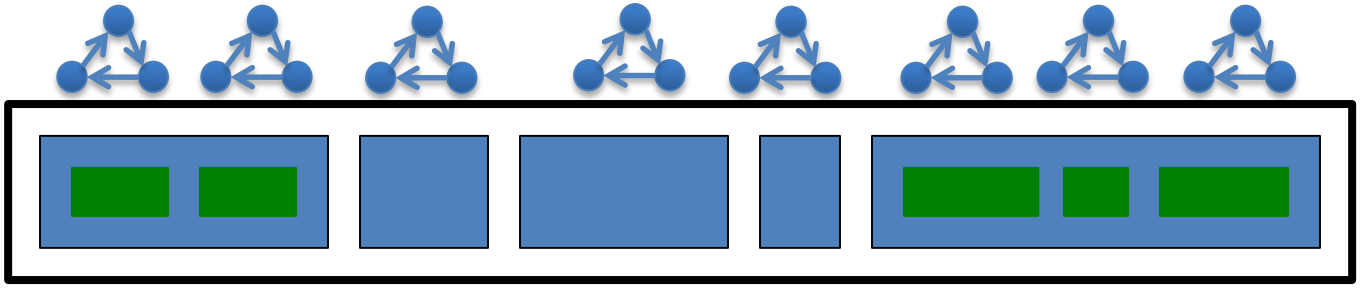


Strategy (cont.)

- Should be able to describe the strategy in a couple of sentences
- Use one strategy per problem
- A strategy is implemented with *tactics*
  - Tasks
  - Ideas
  - Concepts
- Each part of the strategy must be implemented with one or more tactics

Tactics

- Tactics are how you implement the strategy
- Example: Following the line at full speed
  - Implement a good recovery mechanism
  - Make sure your tires have good traction
- Tactics may be composed of multiple simpler tactics
- How do you put it all together?



Program Planning

- Formulate a strategy
  - Convince yourself that you can implement it
  - Identify the tactics you will need
- For each tactic
  - Design a state transition diagram
  - Design corresponding part of the program
- Put the parts together

- How much time will this take?
- How many resources will you need?
- What to do when things go wrong?

How well do You Manage your Project?

- The first project is beginning!
- Do you have enough time for the labs?
- Are you able to get everything done?
- Did you make effective use of your time/resources/money?
- What about dependencies?
- If things go wrong?

Project Management & Team Working

The Problem

- A project consists of many parts
  - Goals and milestones
  - Tasks
  - Resources
  - Dependencies
  - Risks
- To complete a project
  - Accomplish all goals
  - Finish all tasks on time
  - Use only the allocated resources
  - Satisfy all dependencies
  - Adapt to things going wrong

Example: Robotics Project

- Goals
  - Finish the event by Oct 11/12
  - Finish Reports by Oct 13/14
- Tasks
  - Develop a program
  - Write a Report
  - Compete in the event
- Resources
  - One Robot
  - Simulator
  - Three to five people
- Time
  - Two 2-hour lab periods
  - 7 evenings and nights
  - Long weekend
- Dependencies
  - Program must be finished before the Presentation period
  - Report must be completed after the event
  - One Robot?
- Risks? (Plan B)

The Goal of Project Management

- Complete a project on time and on budget

- Identify and schedule tasks
- Allocate resources
- Anticipate and manage risks

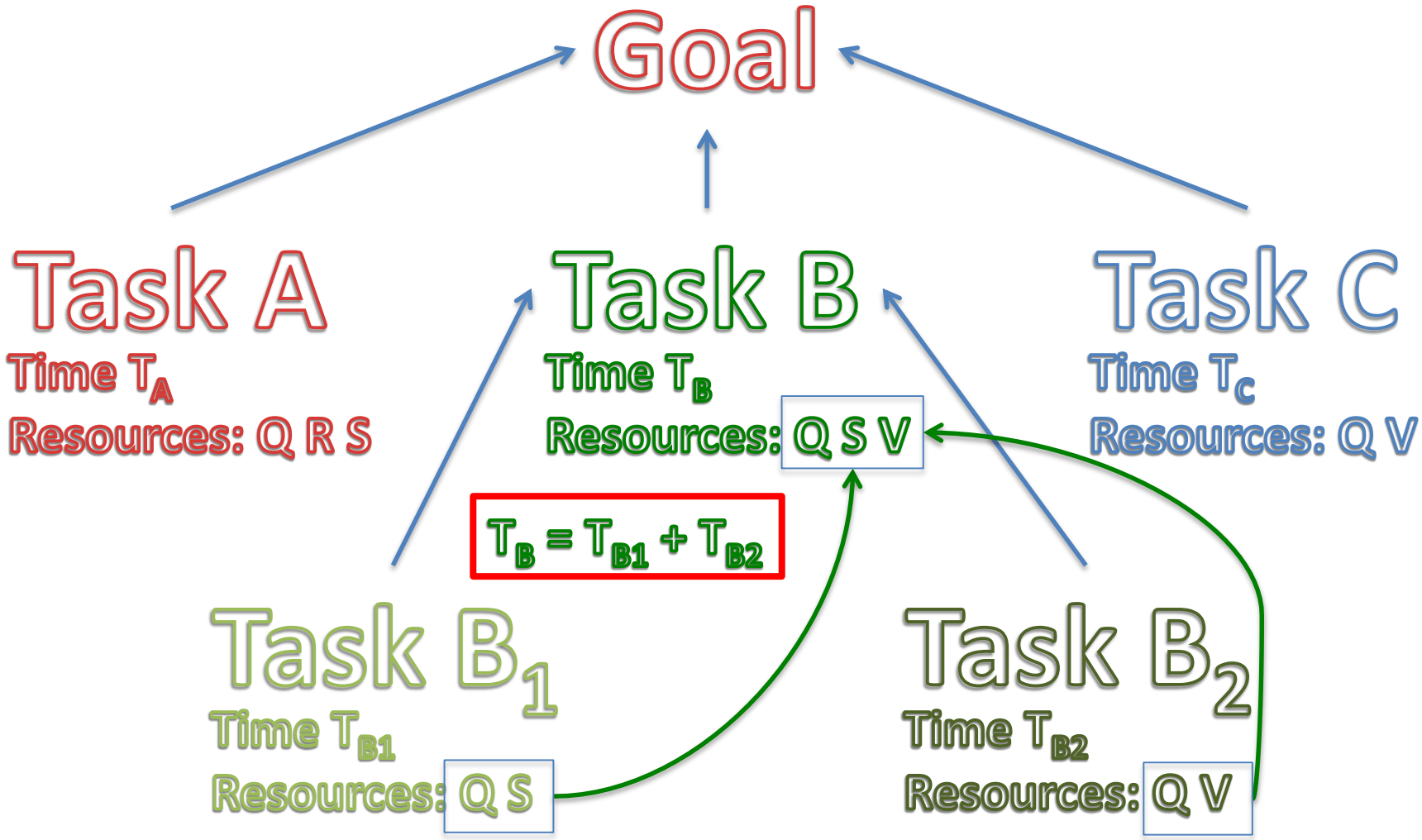
Tasks

- A task
  - Is a piece of work that somebody has to do
  - Takes a minimum amount of time to complete
  - Requires specific resources
  - May Require certain other tasks to be completed first
  - May need to be completed before other tasks can begin
  - **May take longer than expected due to unanticipated events**
- For each task we need to identify
  - What the task is
  - What resources it requires
  - What tasks does it depend on
  - How much time the task will take

Identifying Tasks

- How do we identify all the tasks?
- Idea: Work backwards (reverse engineering)
  - Start with the end goal
  - Ask what task(s) are needed to achieve the goal
  - Ask what resources are needed for the tasks
  - For each task break it down into subtasks and repeat
- Does this sound familiar?

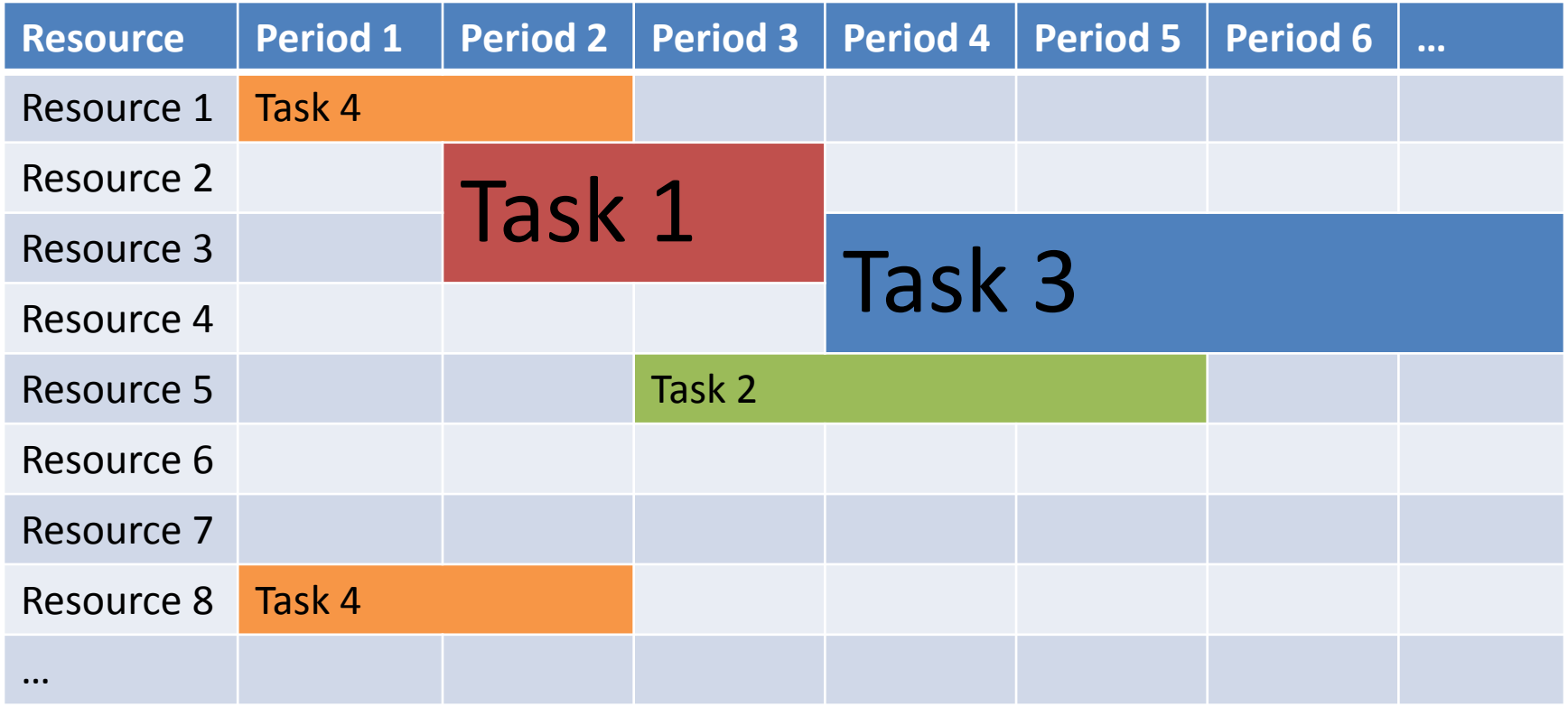
Identifying Tasks



Scheduling Tasks & Allocating Resources

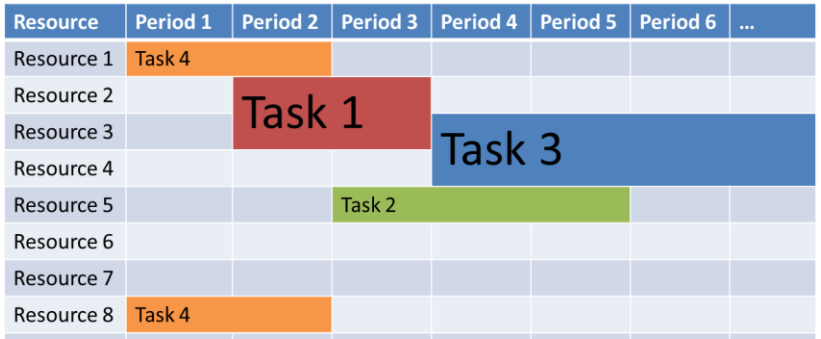
- Problem:
  - There are many tasks
  - There are many resources
  - Each task may have multiple dependencies
- Need to
  - Organize all of them in one place
  - Sort dependencies
  - Check for resource contention (i.e. capabilities)
- Idea: Use a Gantt chart

Gantt Charts



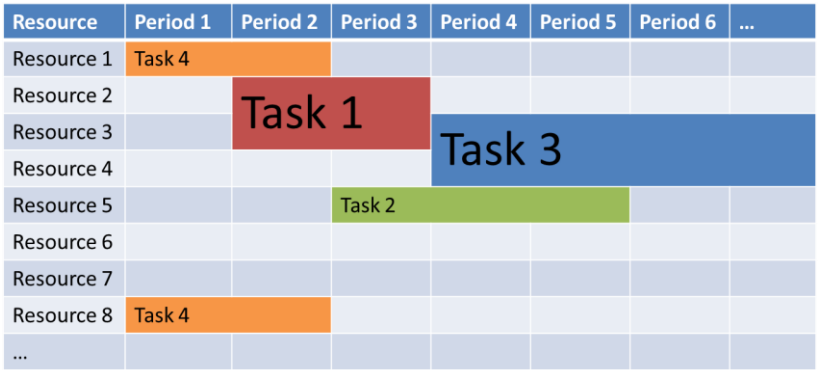
Gantt Chart Rules

- Time is represented horizontally (left to right)
- Resources are denoted vertically
- A task requires both time and resources
  - Represented by one or more rectangles
- If two tasks require the same resource, they cannot overlap
- If task A depends on task B, task A must follow task B
- The minimum amount of time needed to fit in all the tasks is the minimum amount needed for the project



Purpose of Gantt Chart

- Represent all tasks
- Represent resource use
- Represent dependencies
- Represent time of tasks



- Question: How do we know where to place what on the chart?

Three Main Issues

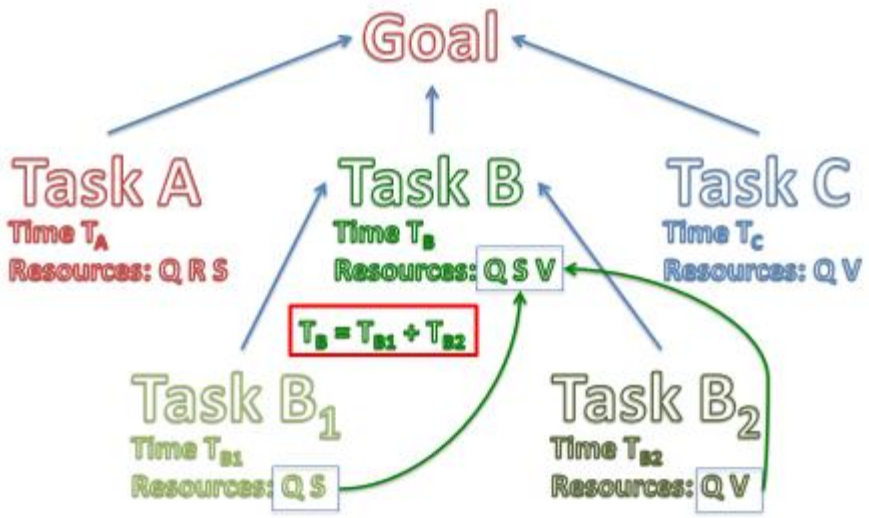
- Dependency chains
- Resource contention
- Risk management

Dependency Chains

- Task A depends on B depends on C depends on D …
- Time of longest chain is the minimum time of the project
- Place longest chain first
- Then the next longest …

Resource Contention

- Tasks cannot use a resource at the same time
- A *bottleneck* occurs when many tasks need the same resource
- Solution:
  - Stagger tasks to avoid resource contention
  - Add more resources to contention



Risk Management

- Things will take longer than you think!
  - What happens to our schedule if we cannot find a solution for software bugs?
- How do we accommodate this fact of life?
- Solutions:
  - Schedule tasks as early as possible to provide time to deal with unforeseen events
  - Schedule extra time for each task
    - 10% to 15% extra time per task is not uncommon