

CS 240: Algorithm Design and Analysis

Kewei Tu & Rui Fan
ShanghaiTech University
Spring 2020

Administrative Stuff

Classes

- 12 weeks (Mar. 2 - late May)

Instructors:

- 屠可伟 tukw@shanghaitech.edu.cn
- 范睿 fanrui@shanghaitech.edu.cn

TA: 贾子夏、李梓玥、陈晋晔、许惟锴、许泗杭、李承谦、王乐童

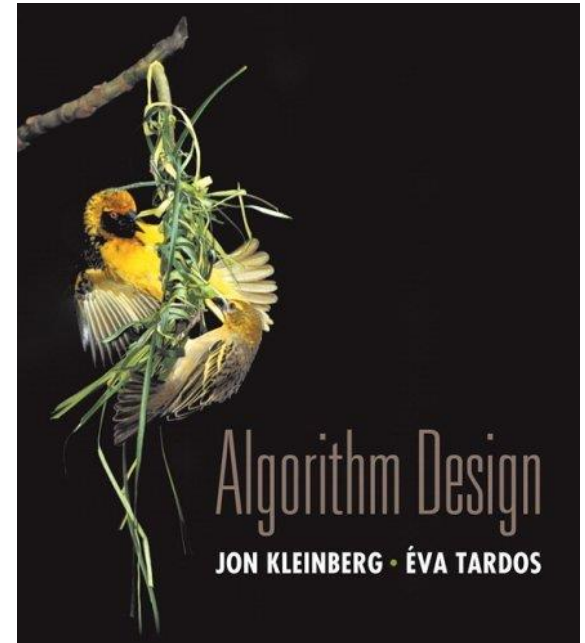
Administrative Stuff

Textbook

- Algorithm Design, Jon Kleinberg and Éva Tardos
- Reference
 - Introduction to Algorithms, C. E. Leiserson, C. Stein, T. H. Cormen, and R. Rivest, (third edition)

Prereq

- Computer Programming
- Data Structures and Algorithms (undergraduate level)



Administrative Stuff

Grading (percentages are tentative)

- Homework (20%): ~5 homework assignments, due in one week
- Midterm (35%): in mid April
 - May be canceled if campus is still closed by then
- Final (35%): in late May or early June
- Project (10%): to be determined

Administrative Stuff

Blackboard (<https://elearning.shanghaitech.edu.cn:8443>)

- Lecture videos, slides
- Announcements
- Homework assignments

Piazza

(<https://piazza.com/shanghaitech.edu.cn/spring2020/cs240/home>)

- QA and discussions

Gradescope

- Homework submission and grading

Administrative Stuff

Online Lectures

- Each lecture video and slide will be uploaded to Blackboard at least one day before the lecture time
- Students watch the video & slide, ideally before or during lecture time
- You can ask questions on Piazza at any time and we will answer them within 24 hours
- Voice chat during lecture time
 - You can request voice chat with us in your Piazza question or reply
 - We will schedule a chat with you during lecture time (1-2:40pm, Mon&Wed)
 - Software: Tencent meeting

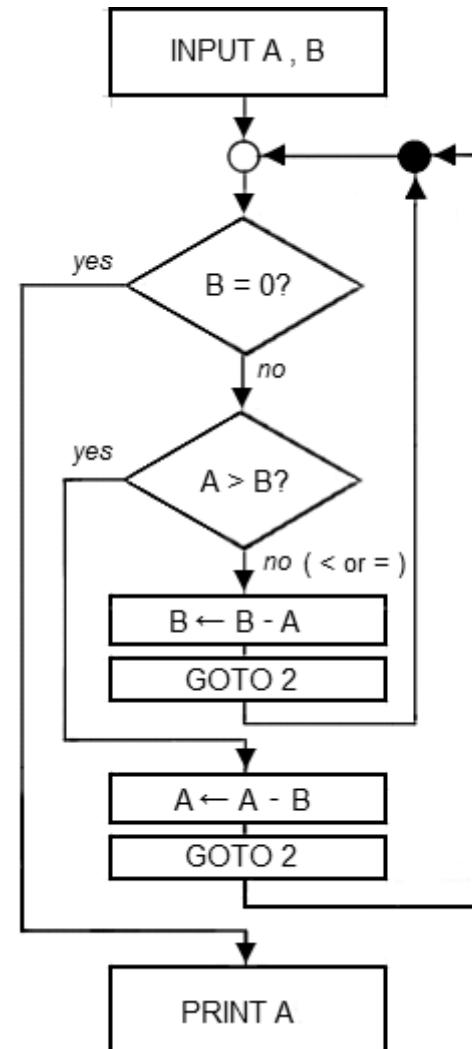
Course Overview

Algorithms

Algorithm.

[Knuth, TAOCP] An algorithm is a finite, definite, effective procedure, with some input and some output.

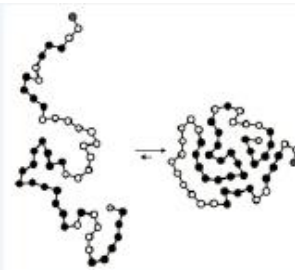
[Wikipedia] An algorithm is a finite sequence of well-defined, computer-implementable instructions, typically to solve a class of problems or to perform a computation.



Why study algorithms?

Wide range of applications.

- **Internet.** Web search, packet routing, distributed file sharing, ...
- **Biology.** Human genome project, protein folding, ...
- **Computers.** Circuit layout, databases, caching, networking, compilers, ...
- **Computer graphics.** Movies, video games, virtual reality, ...
- **Security.** Cell phones, e-commerce, voting machines, ...
- **Multimedia.** MP3, JPG, DivX, HDTV, face recognition, ...
- **Social networks.** Recommendations, news feeds, advertisements, ...
- **Physics.** N-body simulation, particle collision simulation, ...
- ...



Typical Undergraduate Algorithm Course

Understanding and implementing classic algorithms

- Sorting
- Searching
- String algorithms
- Graph algorithms

Critical thinking, problem-solving, coding

This Course

Design and analysis of computer algorithms

- Greedy algorithms
- Divide-and-conquer
- Dynamic programming
- Network flow
- Intractability (complexity classes)
- Coping with intractability
- Approximate algorithms
- Randomized algorithms
- Local search

Critical thinking, problem-solving, rigorous analysis

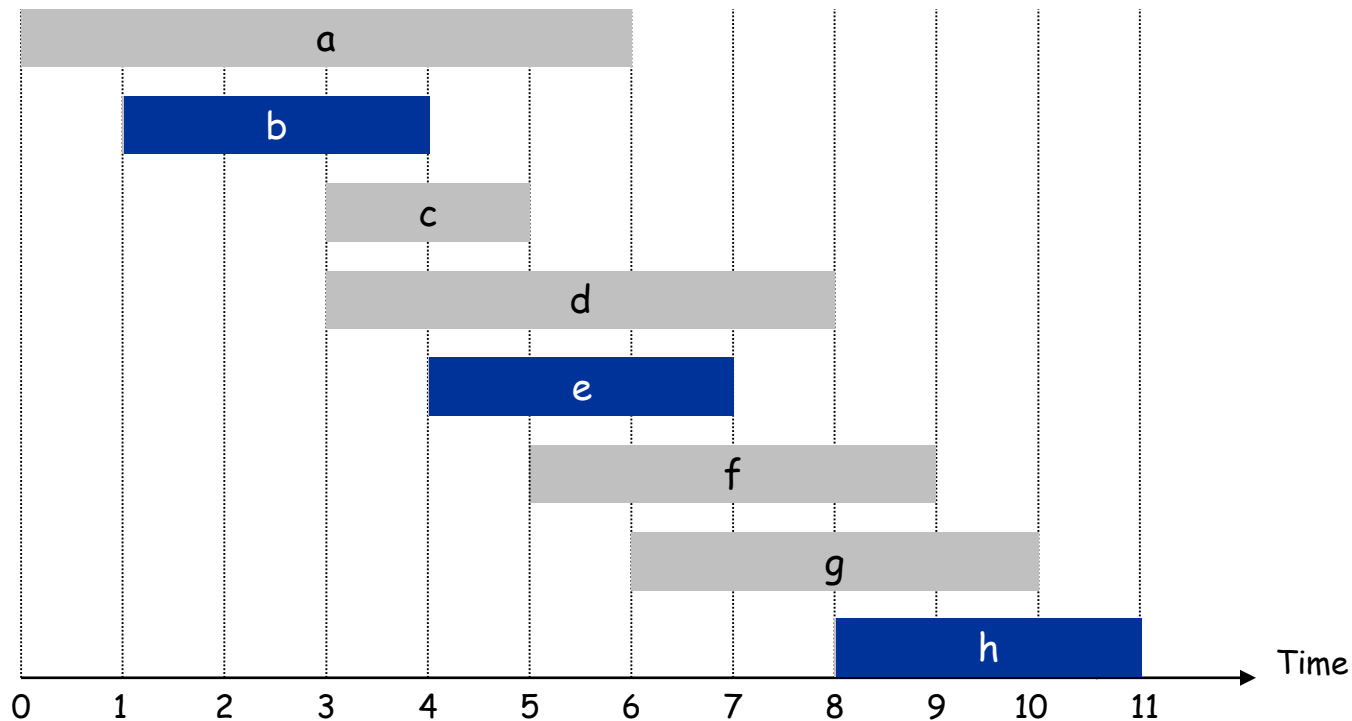
Five Representative Problems

Interval Scheduling

Input. Set of jobs with start times and finish times.

Goal. Find **maximum cardinality** subset of mutually compatible jobs.

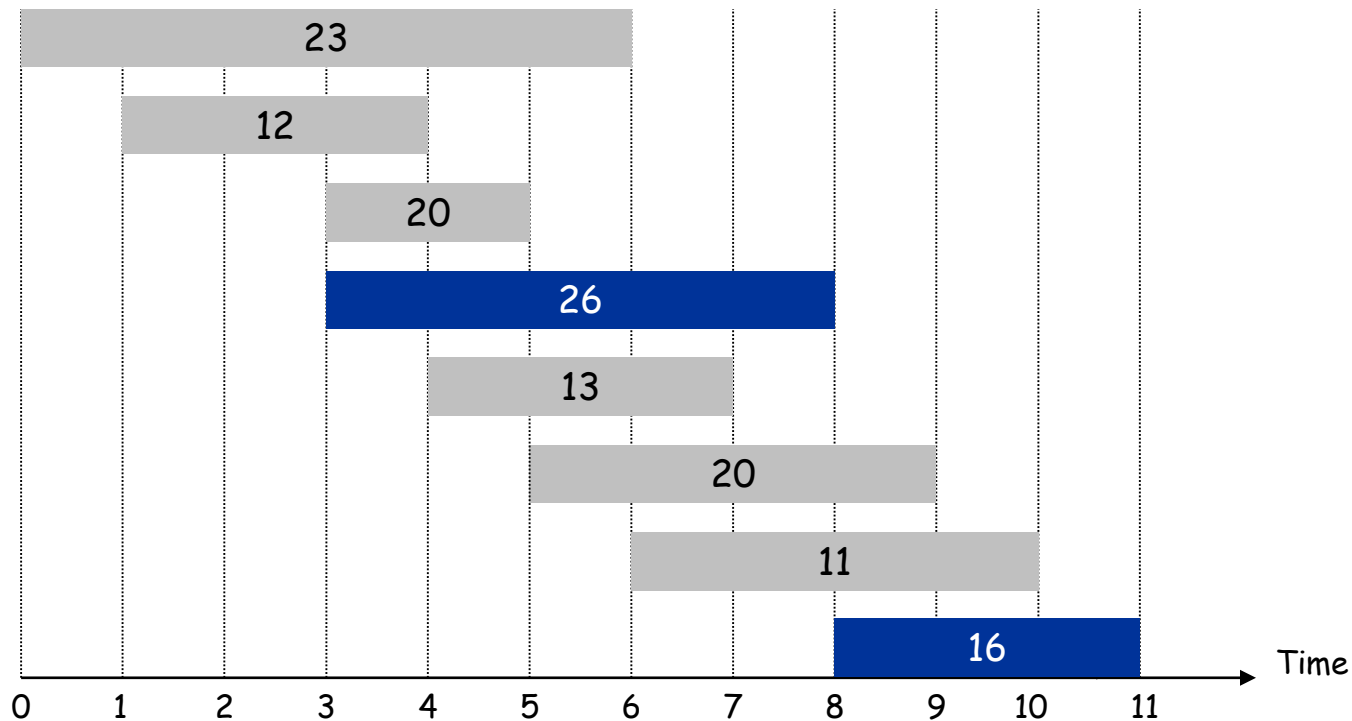
↑
jobs don't overlap



Weighted Interval Scheduling

Input. Set of jobs with start times, finish times, and weights.

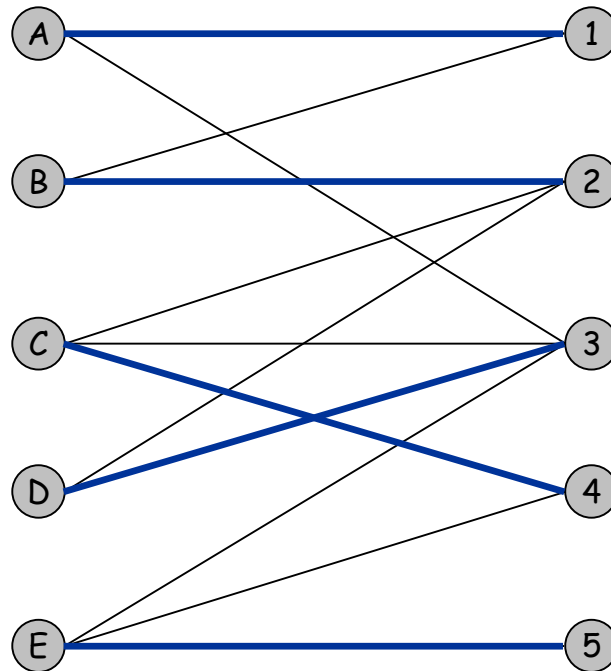
Goal. Find **maximum weight** subset of mutually compatible jobs.



Bipartite Matching

Input. Bipartite graph.

Goal. Find **maximum cardinality** matching.

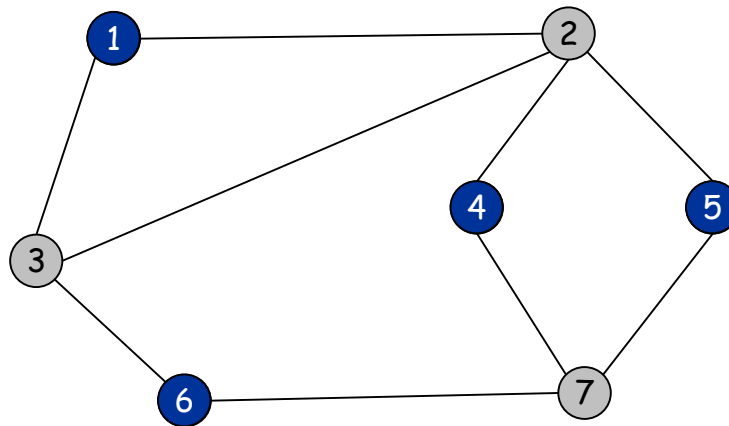


Independent Set

Input. Graph.

Goal. Find **maximum cardinality** independent set.

↑
subset of nodes such that no two
joined by an edge



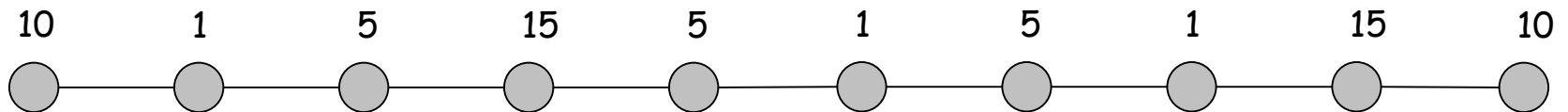
Extension: Weighted independent set.

Competitive Facility Location

Input. Graph with weight on each node.

Game. Two competing players alternate in selecting nodes. Not allowed to select a node if any of its neighbors have been selected.

Goal. Select a **maximum weight** subset of nodes.



Second player can guarantee 20, but not 25.

Five Representative Problems

Variations on a theme: independent set.

Interval scheduling: $n \log n$ greedy algorithm.

Weighted interval scheduling: $n \log n$ dynamic programming algorithm.

Bipartite matching: n^2 max-flow based algorithm.

Independent set: NP-complete.

Competitive facility location: PSPACE-complete.