CS 240: Algorithm Design and Analysis

Kewei Tu & Rui Fan
ShanghaiTech University
Spring 2021

Administrative Stuff

Classes

- 12 weeks (Feb. 23 – mid May)

Instructors:

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

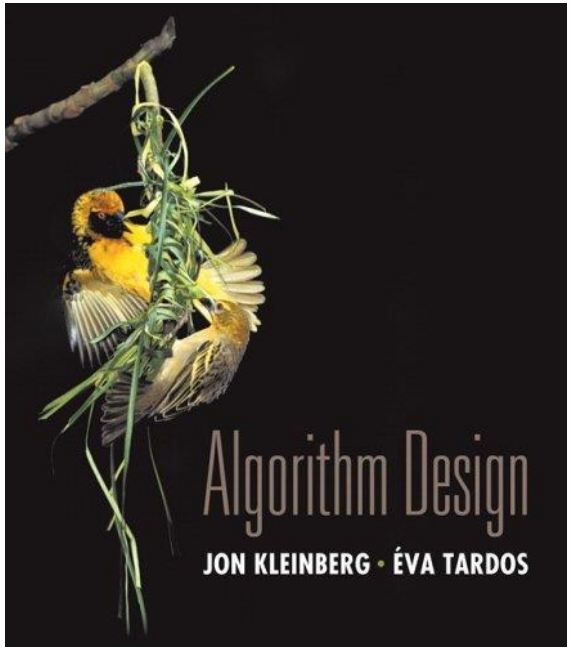
TA: TBD

- Office hours TBD

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

- Homework (20%): ~5 homework assignments, due in one week
- Midterm (35%): possibly in the 7th week
- Final (35%): possibly in mid or late May
- Project (10%): to be determined

- Grading will be curved.
- Undergraduate and graduate students will be graded separately.

Administrative Stuff

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

- Lecture slides
- Announcements
- Homework assignments

Piazza (piazza.com/shanghaitech.edu.cn/spring2021/cs240/home)

- QA and discussions
- Please enroll yourself

Gradescope

- Homework submission and grading

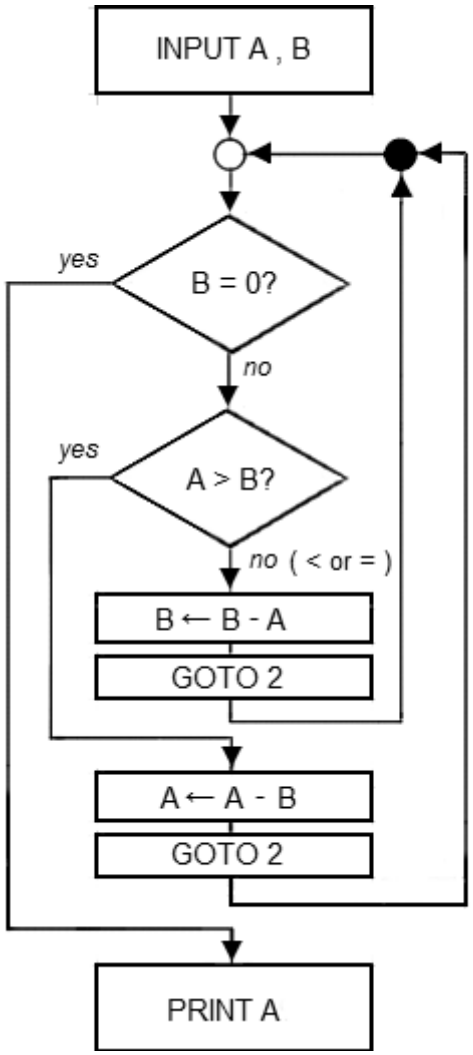
Course Overview

Algorithms

Algorithm.

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

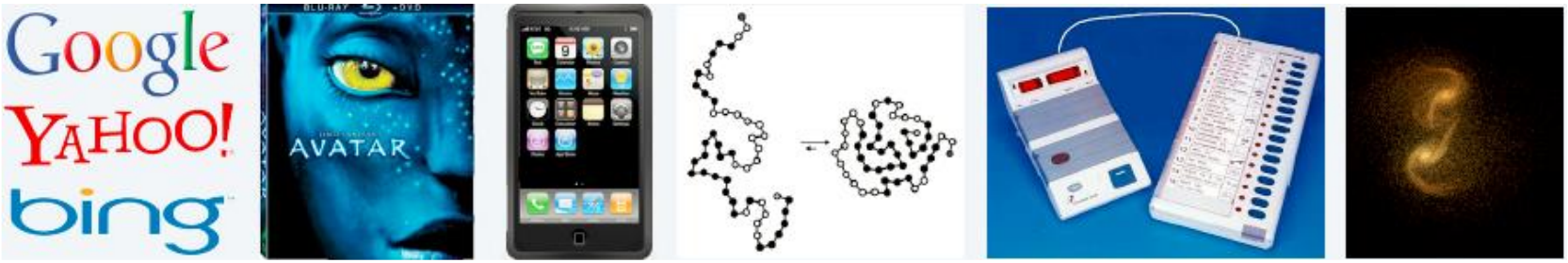
[Wikipedia] An algorithm is a finite sequence of well-defined, computer-implementable instructions, typically to solve a class of problems or to perform a computation.



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*

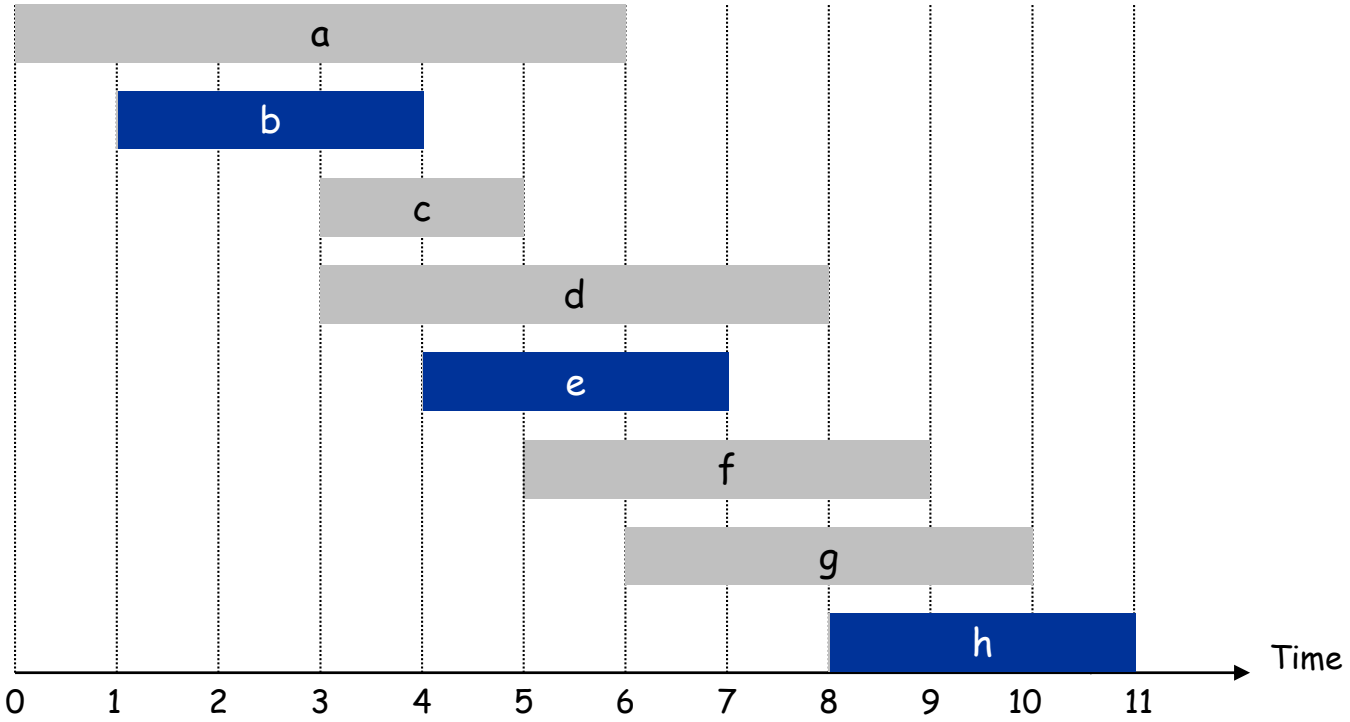
Five Representative Problems

Interval Scheduling

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

Goal. Find maximum cardinality subset of mutually compatible jobs.

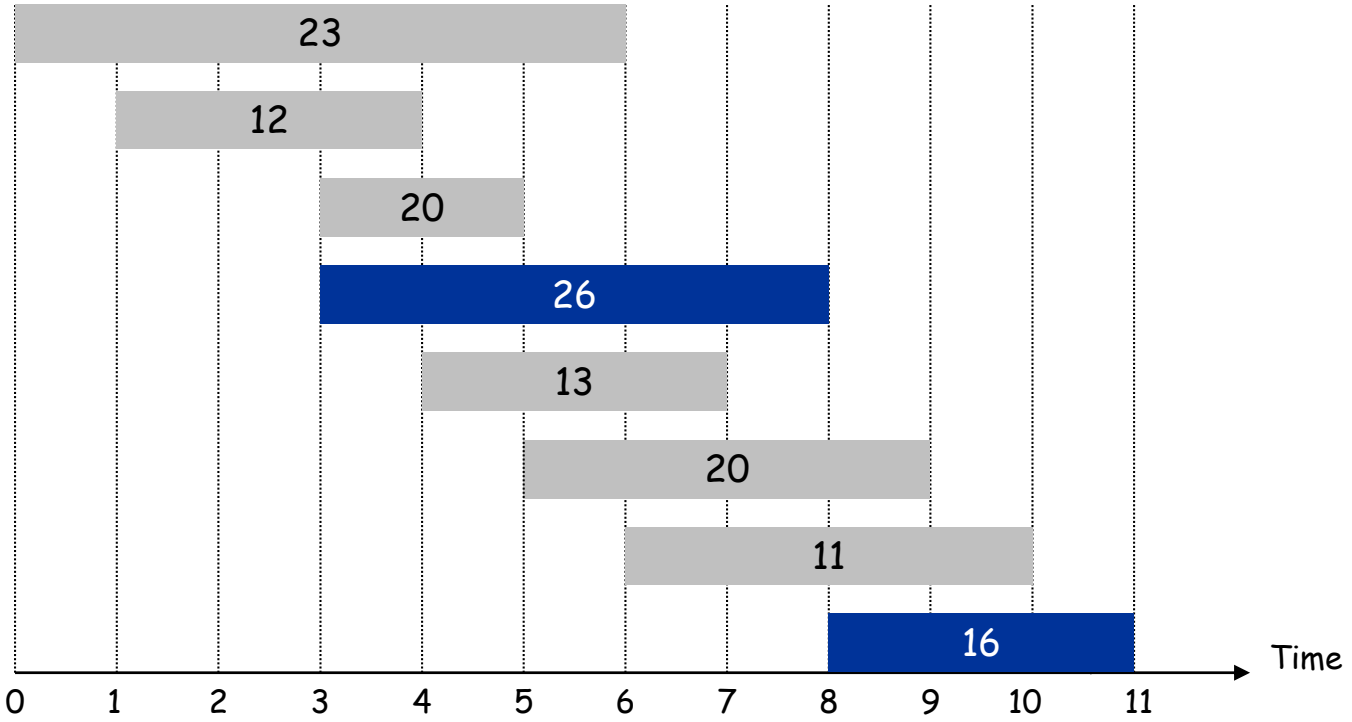
jobs don't overlap



Weighted Interval Scheduling

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

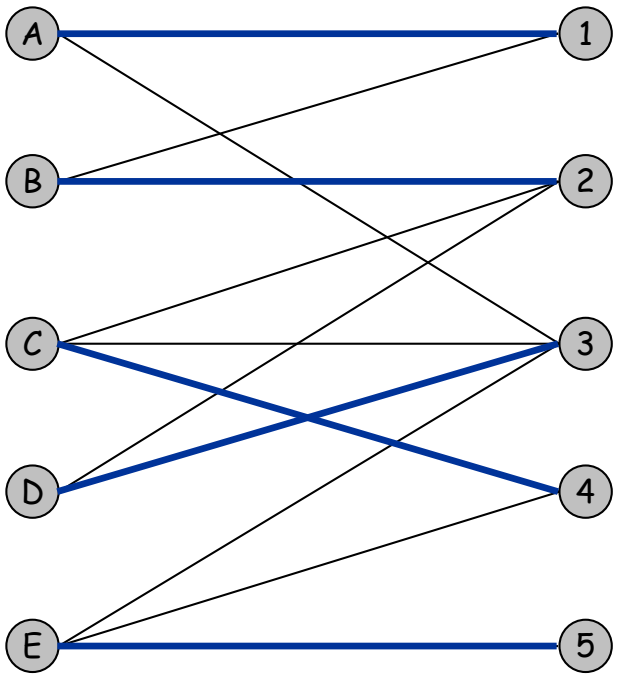
Goal. Find maximum weight subset of mutually compatible jobs.



Bipartite Matching

Input. Bipartite graph.

Goal. Find maximum cardinality matching.

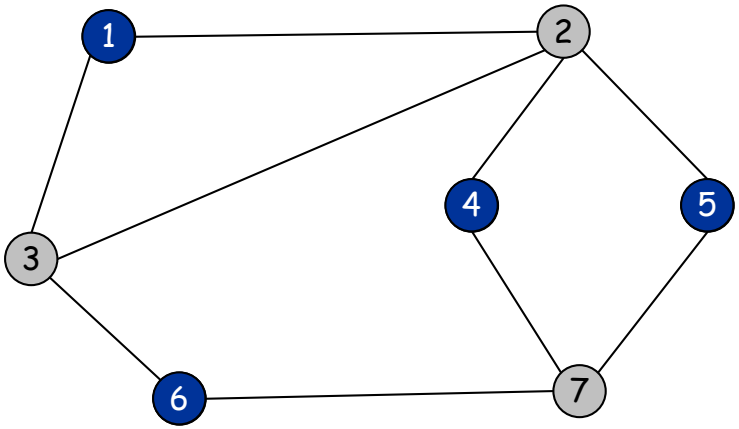


Independent Set

Input. Graph.

Goal. Find maximum cardinality independent set.

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



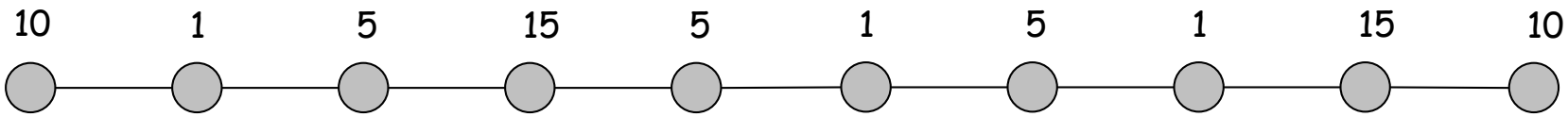
Extension: Weighted independent set.

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.