

Introduction to Algorithms

Chapter 0: Course information

Manuel

Fall 2017

Outline

- 1 Logistics
- 2 Evaluations
- 3 Resources

Who?

Teaching team:

- Instructor: Manuel (charlem@sjtu.edu.cn)
- Teaching assistants:
 - Zhengyang (crukedshfeng@sjtu.edu.cn)
 - Jiaqi (alvin_zuo@sjtu.edu.cn)

Who?

Teaching team:

- Instructor: Manuel (charlem@sjtu.edu.cn)
- Teaching assistants:
 - Zhengyang (crukedshfeng@sjtu.edu.cn)
 - Jiaqi (alvin_zuo@sjtu.edu.cn)

Important notes:

- When contacting a TA for an important matter such as updating a grade cc the message to the instructor
- Add the tag [ve477] to the email subject, e.g.
Subject: [ve477] grade issue
- Do not send large files (> 2 MB) by email, instead use the “Large file upload” assignment on Canvas

When?

Course organisation:

- Lectures:
 - Tuesday 8:00 – 9:40
 - Thursday 8:00 – 9:40
 - Friday 12:10 – 13:50 (even weeks)
- Labs: TBA
- Office hours:
 - Tuesday 14:00 – 15:50
 - Thursday 14:00 – 15:50

Appointments outside of the office hours can be taken by email

What?

Main goals of this course:

- Become familiar with the most common problems and algorithm paradigms
- Understand how to properly analyse and abstract a problem
- Be able to identify or design clear and efficient algorithms to solve a problem

What?

Main goals of this course:

- Become familiar with the most common problems and algorithm paradigms
- Understand how to properly analyse and abstract a problem
- Be able to identify or design clear and efficient algorithms to solve a problem

Ultimate goal: be able to solve a given problem and assess the validity, quality, and efficiency of the solution

How?

Learning strategy:

- Course side:
 - ① Understand the basic concept of algorithmic
 - ② Know the most common problems and their solutions
 - ③ Get an overview of the wide applications of algorithms

How?

Learning strategy:

- Course side:
 - ① Understand the basic concept of algorithmic
 - ② Know the most common problems and their solutions
 - ③ Get an overview of the wide applications of algorithms
- Personal side:
 - ① Read and write code
 - ② Relate known strategies to new problems
 - ③ Perform extra research

Course outcomes

Detailed goals:

- Be able to write clean and clear pseudocode
- Be proficient at using all the basic algorithm paradigms
- Be able to assess the difficulty of a given problem, e.g belongs to P, NP, PSPACE...
- Develop critical thinking abilities
- Know when and how to apply dynamic programming
- Know when and how to apply linear programming
- Have a precise idea of the pros and cons for the most common data structures
- Know how to efficiently solve the most common mathematical problems
- Have a basic idea on how to design multi-threaded algorithms
- Be able to efficiently implement the most common algorithms

Outline

- 1 Logistics
- 2 Evaluations
- 3 Resources

Assignments, labs, and project

Assignments:

- Total: 8
- Content: basic concepts, critical thinking, prove results

Labs:

- Total: 8
- Content: implement common algorithms, learn Python

Project:

- Total: 1, split into three phases
- Content: write a catalog of the most common algorithms

Grading policy

Grade weighting:

- Assignments: 25%
- Projects: 15%
- Labs: 10%
- Midterm exam: 25%
- Final exam: 25%

Grading policy

Grade weighting:

- Assignments: 25%
- Projects: 15%
- Labs: 10%
- Midterm exam: 25%
- Final exam: 25%

Late submission: -10% per day, not accepted after 3 days

Final letter grade: the median will be in the range $\llbracket B, B+ \rrbracket$

L^AT_EX policy

Details of the policy:

- L^AT_EX is a programming language
- Bonus: 10% on the mark if the assignment
 - Is **totally** written in L^AT_EX
 - Is submitted before the deadline
- An assignment grade cannot exceed full mark

L^AT_EX policy

Details of the policy:

- L^AT_EX is a programming language
- Bonus: 10% on the mark if the assignment
 - Is **totally** written in L^AT_EX
 - Is submitted before the deadline
- An assignment grade cannot exceed full mark

An assignment not written in a neat and legible fashion can be deducted up to 10% of the awarded grade

Honor Code

General rules:

- Not allowed:
 - Reuse the code/work from other students
 - Reuse the code/work from the internet
 - Give too many details on how to solve an exercise

Honor Code

General rules:

- Not allowed:
 - Reuse the code/work from other students
 - Reuse the code/work from the internet
 - Give too many details on how to solve an exercise
- Allowed:
 - Reuse part of the code/work from the course/textbooks under the condition of quoting its origin
 - Share ideas and understandings on the course
 - Give hints (not solutions)

Honor Code

Documents allowed during the exams:

- The lecture slides with **notes on them** (paper or electronic)
- A mono or bilingual dictionary

Group works:

- Every student in a group is responsible for his group submission
- If a student breaks the Honor Code, the whole group is sent to Honour Council

Special circumstances

Contact us as early as possible when:

- Facing special circumstances (e.g. full time work, illness. . .)
- Feeling late in the course
- Feeling to work hard without any result

Any late request will be rejected

Outline

- 1 Logistics
- 2 Evaluations
- 3 Resources

On **Canvas** platform:

- Course materials and assignments
- Announcements and notifications
- Polls

References

Places to find information:

- *Algorithm Design*, J. Kleinberg and E. Tardos
- *Introduction to Algorithms*, H. Cormen, C. Leiserson, R. Rivest, and C. Stein
- *The Art of Computer Programming*, D. Knuth
- Search the web

Key points

- Work regularly, do not wait the last minute/day
- Respect the Honor Code
- Go beyond what is taught
- Do not learn, understand
- Keep in touch with us
- Any advice/suggestions will be much appreciated

Thank you!