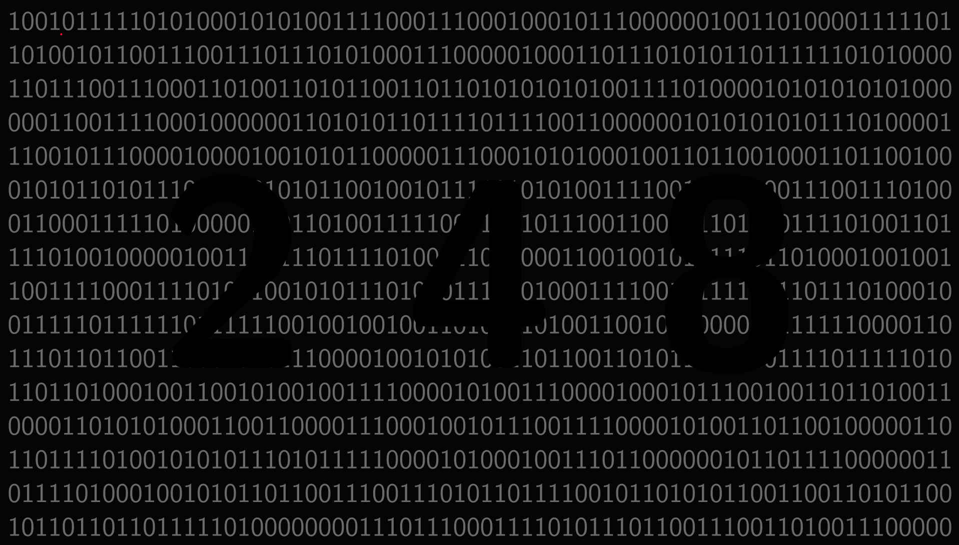
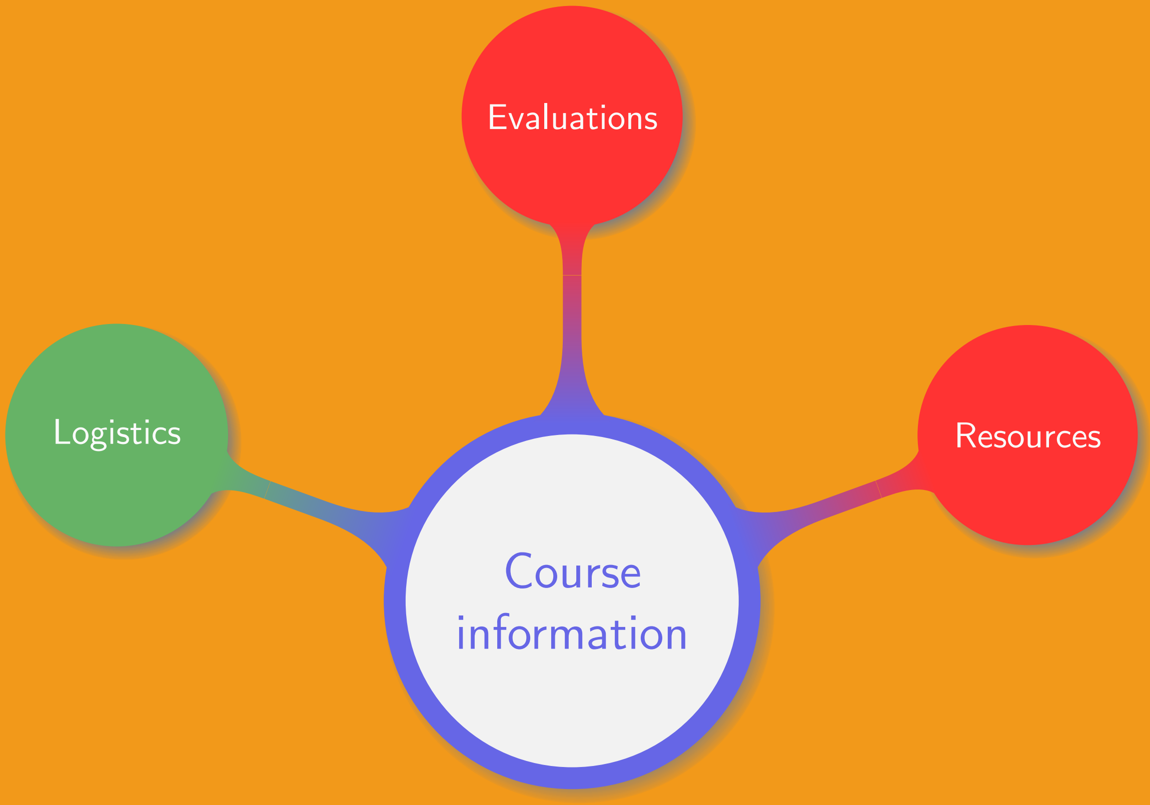


Introduction to Algorithms

0. Course information

Manuel – Fall 2020

Chapter organisation



Course information

Evaluations

Logistics

Resources

Basic information

Teaching team:

- Instructor: Manuel (charlem@sjtu.edu.cn)
- Teaching assistants:
  - Jiayao (jiayaowu1999@sjtu.edu.cn)
  - Yuao (yangyuao@sjtu.edu.cn)

Basic information

Teaching team:

- Instructor: Manuel (charlem@sjtu.edu.cn)
- Teaching assistants:
  - Jiayao (jiayaowu1999@sjtu.edu.cn)
  - Yuao (yangyuao@sjtu.edu.cn)

Important rules:

- When contacting a TA for an important matter, `cc` the instructor
- Prepend [VE477] to the subject, e.g. `Subject: [VE477] Grades`
- Use SJTU jBox service to share large files ($>$ 2 MB)

Basic information

Teaching team:

- Instructor: Manuel (charlem@sjtu.edu.cn)
- Teaching assistants:
  - Jiayao (jiayaowu1999@sjtu.edu.cn)
  - Yuao (yangyuao@sjtu.edu.cn)

Important rules:

- When contacting a TA for an important matter, `CC` the instructor
- Prepend [VE477] to the subject, e.g. `Subject: [VE477] Grades`
- Use SJTU jBox service to share large files ($>$ 2 MB)

Never send large files by email

Course schedule

Course arrangements:

- Lectures:
  - Tuesday 10:00 – 11:40
  - Thursday 10:00 – 11:40
  - Friday 10:00 – 11:40 (even weeks, two lectures only)
- Labs:
  - Thursday 18:20 – 20:30
  - Friday 18:20 – 20:30
- Manuel's office hours:
  - Tuesday 12:15 – 13:45 (JI-437A)
  - Appointment (TBA)
- TAs' office hours: TBA

Course objectives

Main goals of this course:

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

Course objectives

Main goals of this course:

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

Solve a problem, then assess the solution validity, quality, and efficiency

Course workflow

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

Course workflow

Learning strategy:

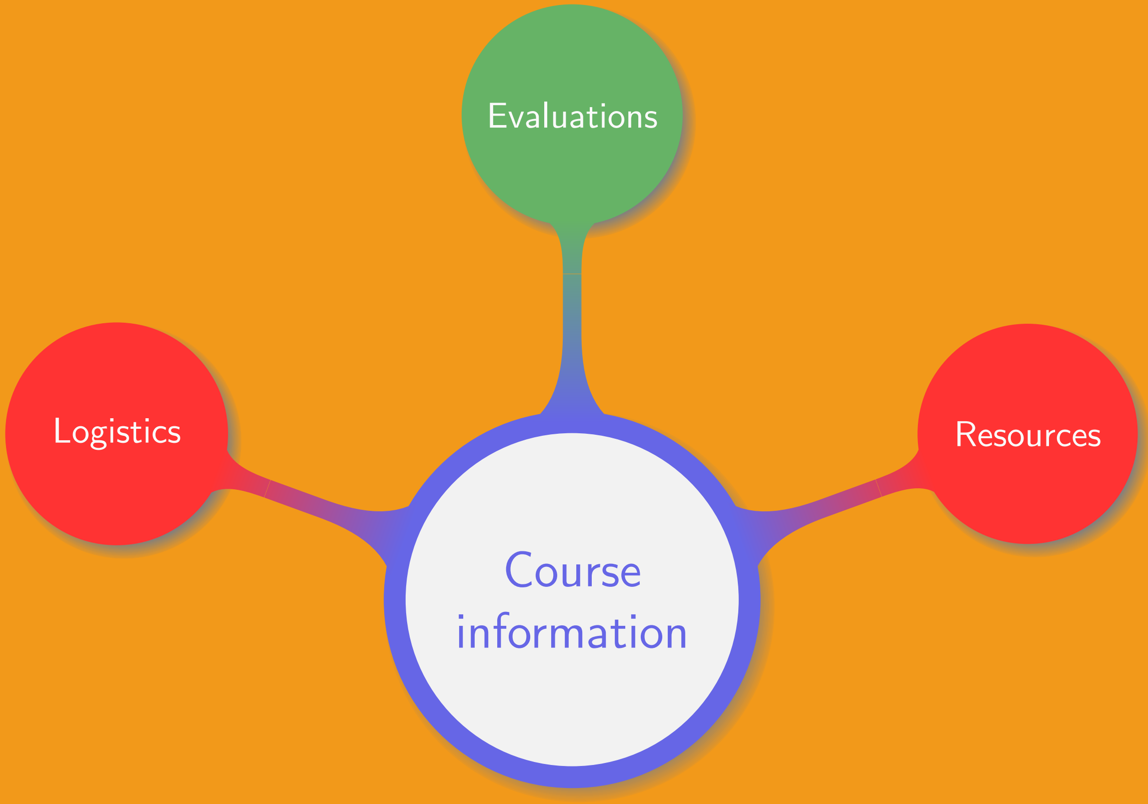
- Course side:
  1. Understand the basic concept of algorithmic
  2. Know the most common problems and their solutions
  3. Get an overview of the wide applications of algorithms
- Personal side:
  1. Read and write code
  2. Relate known strategies to new problems
  3. Perform extra research

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
- Develop critical thinking abilities
- Know when and how to apply dynamic programming
- Have a precise idea of the pros and cons for common data structures
- Know how to efficiently solve common mathematical problems
- Have a basic idea on how to design multi-threaded algorithms
- Be able to efficiently implement the most common algorithms

Chapter organisation



Assignments

Homework:

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

Labs:

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

Project:

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

Challenge: TBA

Grading policy

Grade weighting:

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

Grading policy

Grade weighting:

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

Assignment submissions:

- Bonus: +10% for a work fully written in LaTeX, bounded to 100%
- Penalty: −10% for a work not written in a neat and legible fashion
- Late policy: −10% per day, not accepted after three days

Grading policy

Grade weighting:

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

Assignment submissions:

- Bonus: $+10\%$ for a work fully written in LaTeX, bounded to 100%
- Penalty: -10% for a work not written in a neat and legible fashion
- Late policy: -10% per day, not accepted after three days

Grades will be curved with the median in the range $\llbracket B, B+ \rrbracket$

Honor Code

General rules:

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

Honor Code

General rules:

- Not allowed:
  - Reuse the code or work from other students
  - Reuse the code or work from the internet
  - Give too many details on how to solve an exercise
- Allowed:
  - Share ideas and understandings on the course
  - Provide general directions on where or how to find information

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, etc.
- Feeling late in the course
- Feeling to work hard without any result

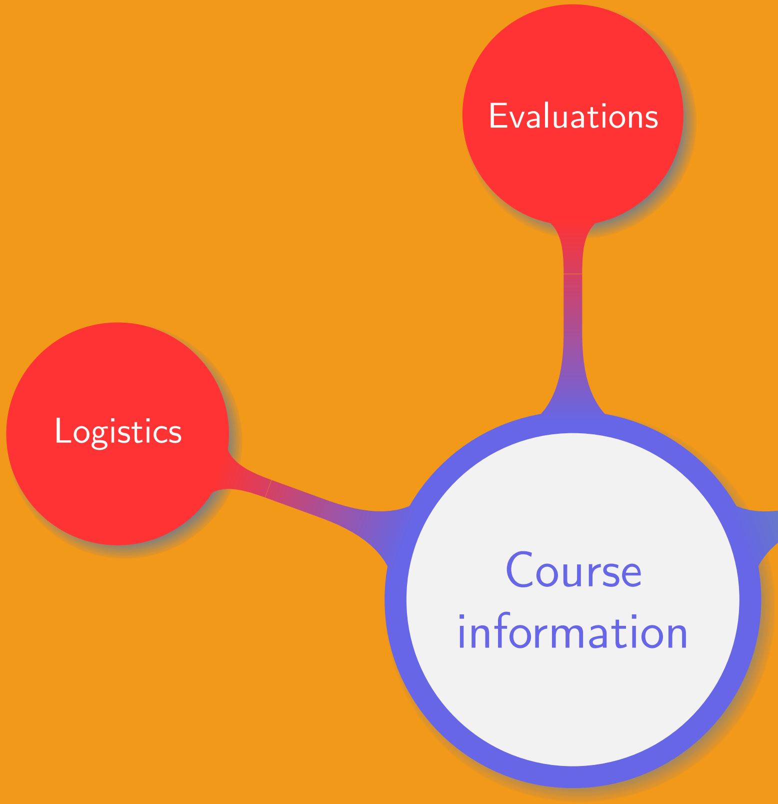
Special circumstances

Contact us as early as possible when:

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

Any late request will be rejected

Chapter organisation



Canvas

On Canvas platform:

- Course materials:
  - Syllabus
  - Lecture slides
  - Homework
  - Labs
  - Projects
  - Challenges
- Course information:
  - Announcements
  - Notifications
  - Grades
  - Surveys

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
- Piazza
- Search information online, i.e. $\{\mathit{internet} \setminus \{\mathit{non\text{-}English\ websites}\}\}$

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
- Piazza
- Search information online, i.e. $\{\textit{internet} \setminus \{\textit{non-English websites}\}\}$

Never use Baidu in any course

Key points

- Work regularly, do not wait the last minute
- Respect the Honor Code
- Go beyond what is taught
- Do not learn, understand
- Keep in touch with us
- Advice and suggestions are always much appreciated

Thank you!