

Teaching team:

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

### Teaching team:

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

### Teaching team:

- Instructor: Manuel ([charlem@sjtu.edu.cn](mailto:charlem@sjtu.edu.cn))
- Teaching assistants:
  - Jiayao ([jiayaowu1999@sjtu.edu.cn](mailto:jiayaowu1999@sjtu.edu.cn))
  - Yuao ([yangyuao@sjtu.edu.cn](mailto: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 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 – 20:20
  - Friday 18:20 – 20:20
- Manuel's office hours:
  - Tuesday 12:15 – 13:45 (JI-437A)
  - Appointment (TBA)
- TAs' office hours: TBA

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

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*



## Learning strategy:

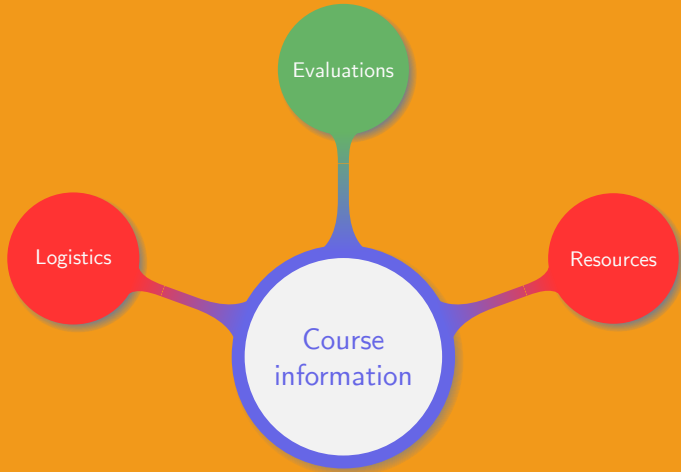
- Course side:
  - 1 Understand the basic concept of algorithmic
  - 2 Know the most common problems and their solutions
  - 3 Get an overview of the wide applications of algorithms

## 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

### 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



### 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

### Grade weighting:

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

### Grade weighting:

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

### Assignment submissions:

- Bonus: +10% for a work fully written in  $\text{\LaTeX}$ , 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

### Grade weighting:

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

### Assignment submissions:

- Bonus: +10% for a work fully written in  $\text{\LaTeX}$ , 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  $\llbracket B, B+ \rrbracket$*



## 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

## 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

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

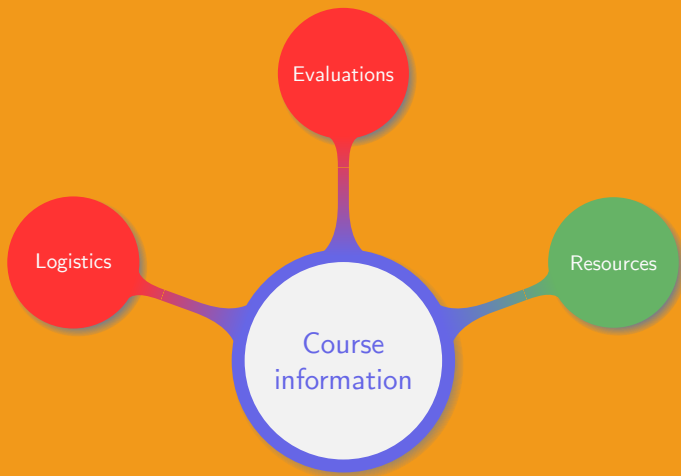
Contact us as early as possible when:

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

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



On **Canvas** platform:

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

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.  $\{internet \setminus \{non-English\ websites\}\}$



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.  $\{internet \setminus \{non-English\ websites\}\}$

Never use Baidu in any course



1001011111010100010101001111000111000100010111000000100110100001111101  
1010010110011100111011101010001110000010001101110101011011111101010000  
11011100111000110100110101100110110101010100111101000010101010101000  
00011001111000100000011010101101111011110011000000101010101110100001  
1100101110000100001001010110000011100010101000100110110010001101100100  
0101011010111000001010110010010111000001010011110000001100111001110100  
01100011111010000001101001111100000110111001100110111101001101  
111010010000010011111011110100110000001100110010010011111010001001001  
10011110001111101001010111010011111010001111001111101110100010  
01111101111110011111001001001001100111001100100100001111110000110  
1110110110011111000010010101001101100110101111110111111010  
11011010001001100101001001111100001010011100001000101110010011011010011  
0000110101010001100110000111000100101110011110000101001101100100000110  
1101111010010101011110101111100001010001001110110000001011011110000011  
0111101000100101011011001110011101011011110010110101011001100110101100  
10110110110111110100000000111011100011110101110110011100111010011100000

Thank you!