极客大学算法训练营
第二课
训练准备和复杂度分析

覃超

Sophon Tech 创始人，前 Facebook 工程师

目录

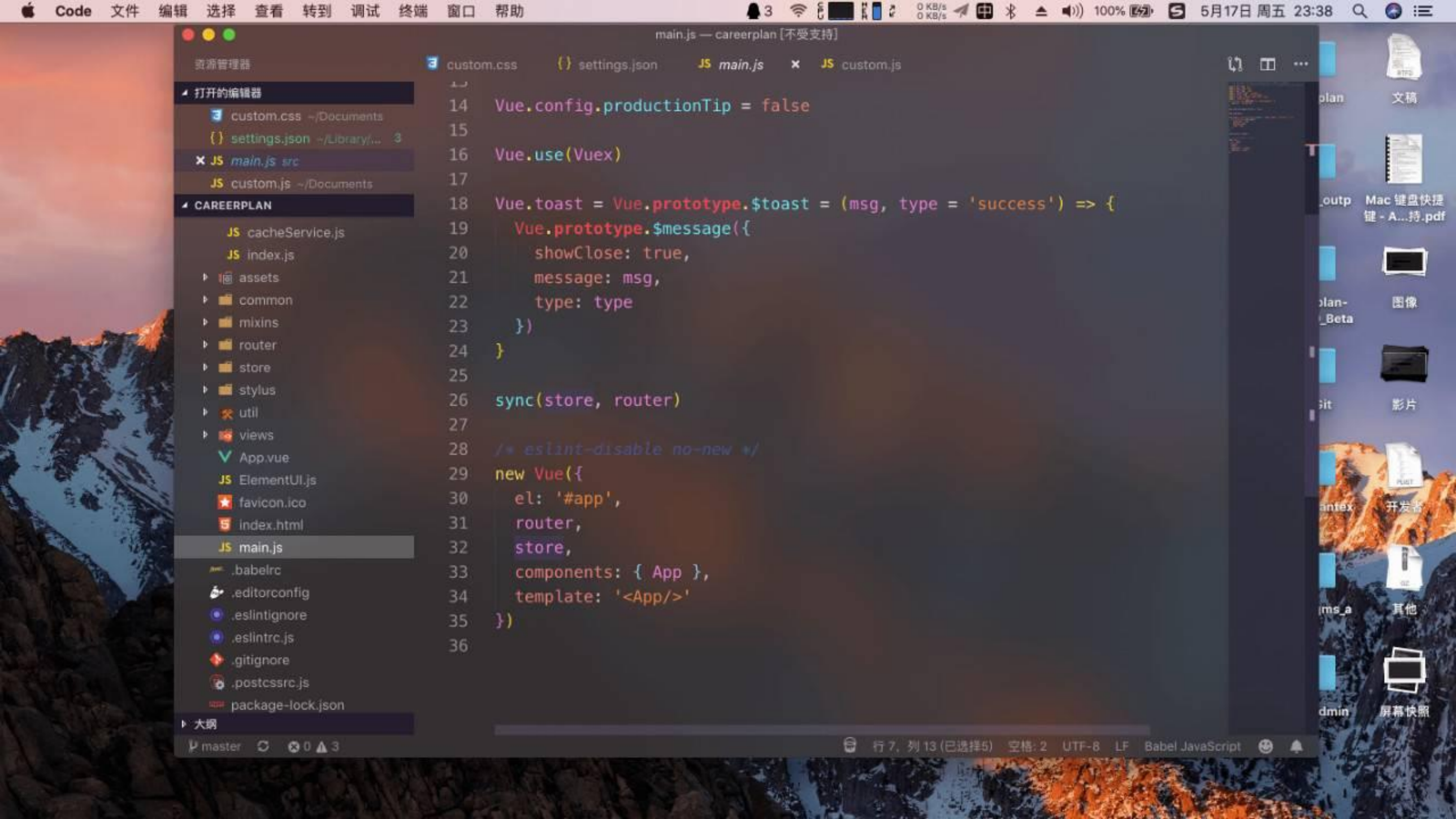
第一节

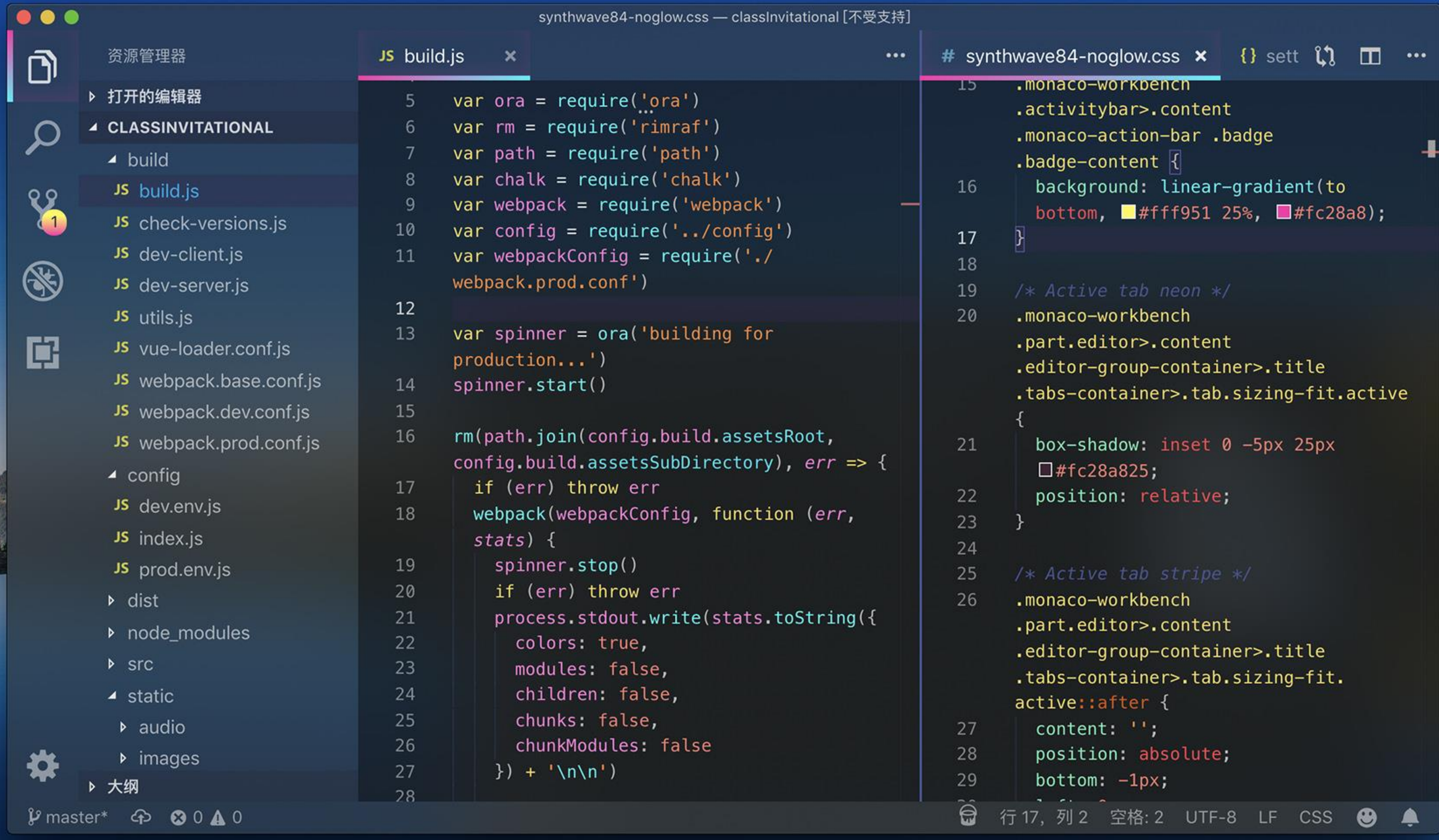
训练环境设置、编码技巧和 Code Style

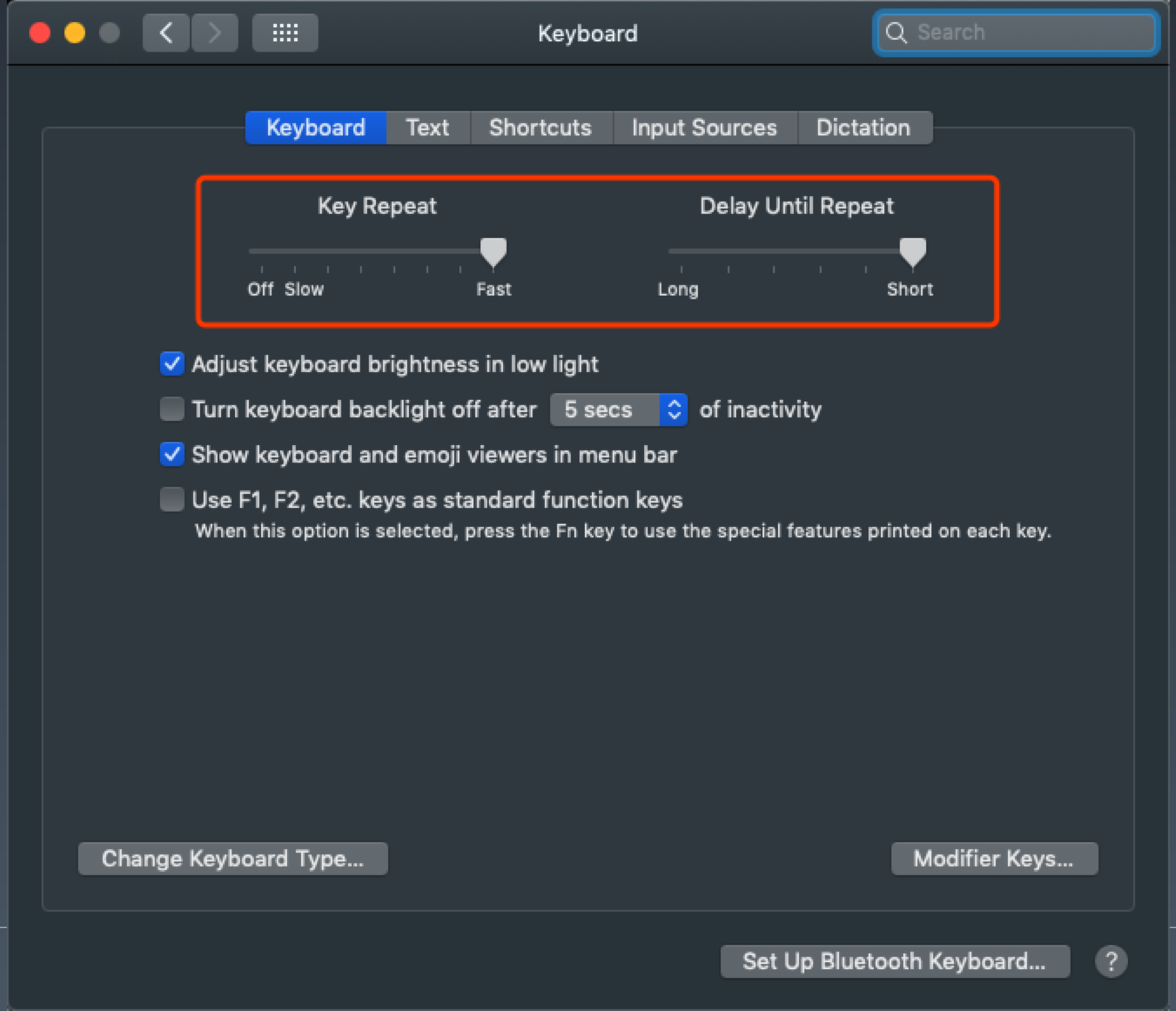
工欲善其事，必先利其器

电脑设置

- Google
- Mac: iTerm2 + zsh (oh my zsh)
  Windows: Microsoft new terminal:
  (https://github.com/microsoft/terminal)
- VSCode; Java: IntelliJ; Python: Pycharm
- LeetCode plugin (VSCode & IntelliJ)
- https://vscodethemes.com/
- 骚操作:
  https://juejin.im/entry/587e0f2f570c352201113e14
  https://juejin.im/post/5ce136515188252 5ff28ed47


Code 文件 编辑 选择 查看 转到 调试 终端 窗口 帮助
5月17日 周五 23:38
main.js — careerplan [不受支持]
资源管理器
打开的编辑器
custom.css ~/Documents
settings.json ~/Library/...
main.js src
custom.js ~/Documents
CAREERPLAN
cacheService.js
index.js
assets
common
mixins
router
store
stylus
util
views
App.vue
ElementUI.js
favicon.ico
index.html
main.js
.babelrc
.editorconfig
.eslintignore
.eslintrc.js
.gitignore
.postcssrc.js
package-lock.json
大纲
custom.css
settings.json
main.js
custom.js
14 Vue.config.productionTip = false
15
16 Vue.use(Vuex)
17
18 Vue.toast = Vue.prototype.$toast = (msg, type = 'success') => {
19 Vue.prototype.$message({
20 showClose: true,
21 message: msg,
22 type: type
23 })
24 }
25
26 sync(store, router)
27
28 /* eslint-disable no-new */
29 new Vue({
30 el: '#app',
31 router,
32 store,
33 components: { App },
34 template: '<App/>'
35 })
36
master
0 3
行 7，列 13 (已选择5)
空格: 2
UTF-8
LF
Babel JavaScript
文稿
Mac 键盘快捷键 - A...持.pdf
图像
影片
开发者
其他
屏幕快照


synthwave84-noglow.css — classInvitational [不受支持]
资源管理器
打开的编辑器
CLASSINVITATIONAL
build
build.js
check-versions.js
dev-client.js
dev-server.js
utils.js
vue-loader.conf.js
webpack.base.conf.js
webpack.dev.conf.js
webpack.prod.conf.js
config
dev.env.js
index.js
prod.env.js
dist
node_modules
src
static
audio
images
大纲
build.js
var ora = require('ora')
var rm = require('rimraf')
var path = require('path')
var chalk = require('chalk')
var webpack = require('webpack')
var config = require('../config')
var webpackConfig = require('./webpack.prod.conf')
var spinner = ora('building for production...')
spinner.start()
rm(path.join(config.build.assetsRoot, config.build.assetsSubDirectory), err => {
if (err) throw err
webpack(webpackConfig, function (err, stats) {
spinner.stop()
if (err) throw err
process.stdout.write(stats.toString({
colors: true,
modules: false,
children: false,
chunks: false,
chunkModules: false
}) + '\n\n')
synthwave84-noglow.css
.monaco-workbench .activitybar>.content .monaco-action-bar .badge .badge-content {
background: linear-gradient(to bottom, #fff951 25%, #fc28a8);
}
/* Active tab neon */
.monaco-workbench .part.editor>.content .editor-group-container>.title .tabs-container>.tab.sizing-fit.active {
box-shadow: inset 0 -5px 25px #fc28a825;
position: relative;
}
/* Active tab stripe */
.monaco-workbench .part.editor>.content .editor-group-container>.title .tabs-container>.tab.sizing-fit.active::after {
content: '';
position: absolute;
bottom: -1px;
master*
0 0
行 17，列 2
空格: 2
UTF-8
LF
CSS


Keyboard
Search
Keyboard
Text
Shortcuts
Input Sources
Dictation
Key Repeat
Off
Slow
Fast
Delay Until Repeat
Long
Short
Adjust keyboard brightness in low light
Turn keyboard backlight off after
5 secs
of inactivity
Show keyboard and emoji viewers in menu bar
Use F1, F2, etc. keys as standard function keys
When this option is selected, press the Fn key to use the special features printed on each key.
Change Keyboard Type...
Modifier Keys...
Set Up Bluetooth Keyboard...
?

Code Style

Java、Python、...

- Google code style
- Facebook
- Airbnb

LeetCode

- leetcode-cn.com 和 题解
- leetcode.com 和 Discuss board

指法和小操作

- home, end（行头、行尾）
- Word 单词、选单词、选整行
- IDE 的自动补全
- Top tips for <IDE-name>

自顶向下的编程方式

- https://markhneedham.com/blog/2008/09/15/clean-code-book-review/
- https://leetcode-cn.com/problems/valid-palindrome/

第二节

时间复杂度、空间复杂度

Big O notation

O(1): Constant Complexity 常数复杂度

O(log n): Logarithmic Complexity 对数复杂度

O(n): Linear Complexity 线性时间复杂度

O(n^2): N square Complexity 平方

O(n^3): N cubic Complexity 立方

O(2^n): Exponential Growth 指数

O(n!): Factorial 阶乘

注意：只看最高复杂度的运算

Big O notation

O(1)

```
int n = 1000;
System.out.println("Hey - your input is: " + n);
```

O(?)

```
int n = 1000;
System.out.println("Hey - your input is: " + n);
System.out.println("Hmm.. I'm doing more stuff with: " + n);
System.out.println("And more: " + n);
```

Big O notation

O(N)

```
for (int i = 1; i <= n; i++) {
        System.out.println("Hey - I'm busy looking at: " + i);
}
```

O(N^2)

```
for (int i = 1; i <= n; i++) {
    for (int j = 1; j <=n; j++) {
        System.out.println("Hey - I'm busy looking at: " + i + " and " +
j);
    }
}
```

Big O notation

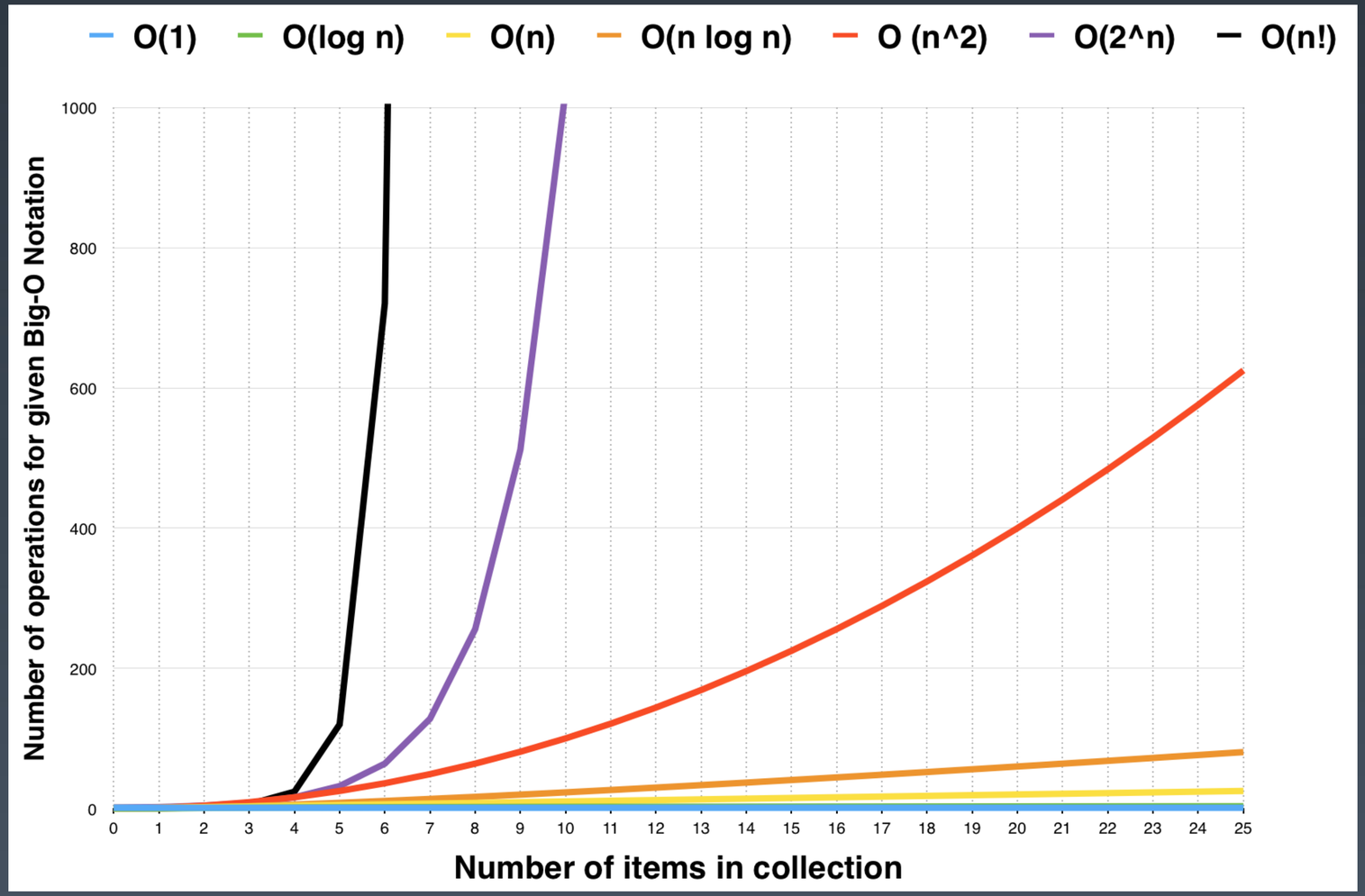
O(log(n))

```
for (int i = 1; i < n; i = i * 2) {
    System.out.println("Hey - I'm busy looking at: " + i);
}
```

O(k^n)

```
int fib(int n) {
    if (n < 2) return n;
    return fib(n - 1) + fib(n - 2);
}
```

时间复杂度曲线



计算：1 + 2 + 3 + … + n

- 方法一：从1到n的循环累加

```
y = 0
for i = 1 to n:
    y += i
```

- 方法二：求和公式 sum = n(n+1)/2

```
y = n * (n + 1) / 2
```

更复杂的情况：递归

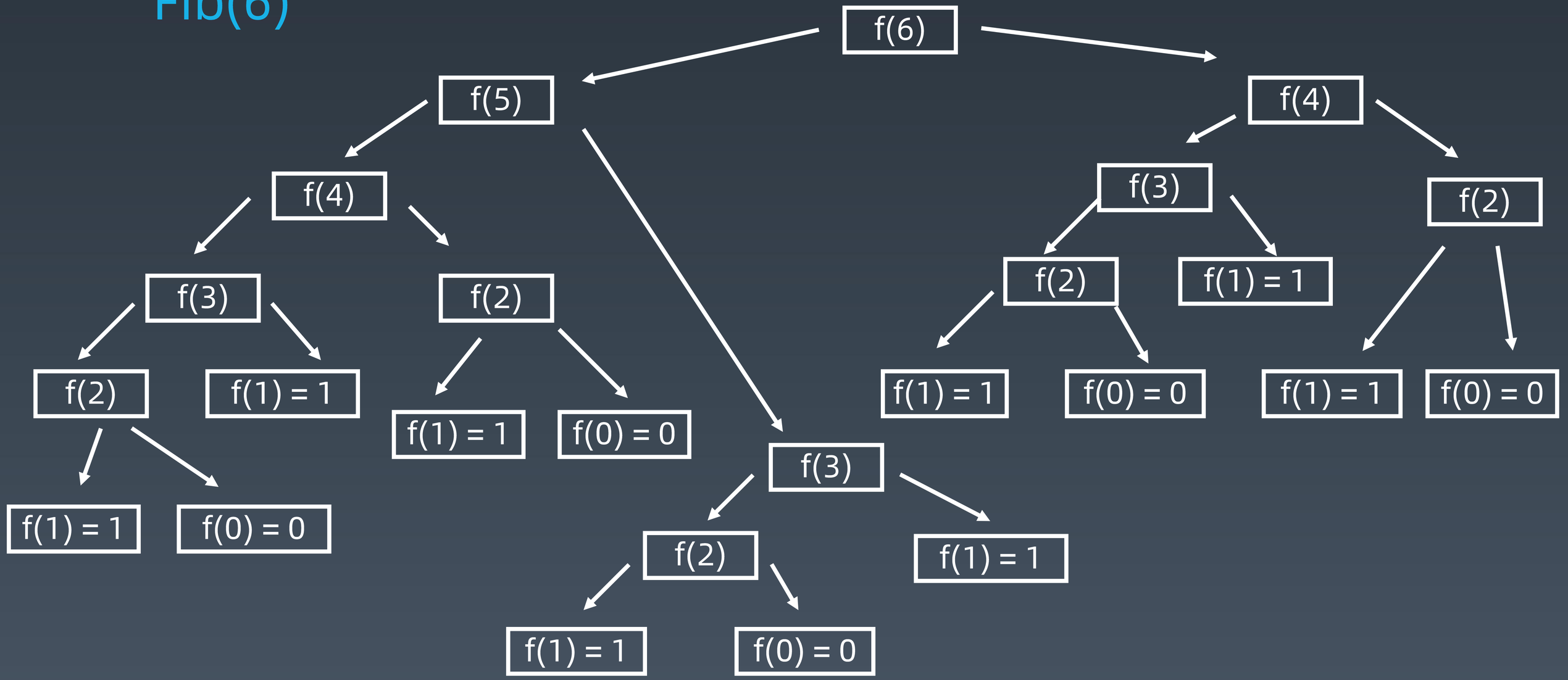
Fib: 0, 1, 1, 2, 3, 5, 8, 13, 21, ...

- F(n) = F(n - 1) + F(n - 2)

- 面试 （直接用递归）

```
int fib(int n) {
    if (n < 2) return n;
    return fib(n - 1) + fib(n - 2);
 }
```

Fib(6)


f(6)
f(5)
f(4)
f(4)
f(3)
f(3)
f(2)
f(2)
f(1) = 1
f(2)
f(1) = 1
f(0) = 0
f(1) = 1
f(0) = 0
f(3)
f(2)
f(1) = 1
f(1) = 1
f(0) = 0
f(1) = 1
f(0) = 0
f(1) = 1
f(0) = 0

Master Theorem

Application to common algorithms [edit]

Algorithm	Recurrence relationship	Run time	Comment
Binary search	$T(n) = T\left(\frac{n}{2}\right) + O(1)$	$O(\log n)$	Apply Master theorem case $c = \log_b a$, where $a = 1, b = 2, c = 0, k = 0$[5]
Binary tree traversal	$T(n) = 2T\left(\frac{n}{2}\right) + O(1)$	$O(n)$	Apply Master theorem case $c < \log_b a$ where $a = 2, b = 2, c = 0$[5]
Optimal sorted matrix search	$T(n) = 2T\left(\frac{n}{2}\right) + O(\log n)$	$O(n)$	Apply the Akra–Bazzi theorem for $p = 1$ and $g(u) = \log(u)$ to get $\Theta(2n - \log n)$
Merge sort	$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$	$O(n \log n)$	Apply Master theorem case $c = \log_b a$, where $a = 2, b = 2, c = 1, k = 0$

思考题

二叉树遍历 - 前序、中序、后序：时间复杂度是多少？

图的遍历：时间复杂度是多少？

搜索算法：DFS、BFS 时间复杂度是多少？

二分查找：时间复杂度是多少？

思考题

二叉树遍历 - 前序、中序、后序：O(N)

图的遍历：O(N)

搜索算法：DFS、BFS - O(N)

二分查找：O(logN)

空间复杂度

实战情况

1. 数组的长度

2. 递归的深度（特殊说明）

实例分析：
https://leetcode-cn.com/problems/climbing-stairs/solution/pa-lou-ti-by-leetcode/

小结

- 常用工具配置
- 基本功和编程指法
- 常见的时间、空间复杂度

THANKS!