



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

Introduction to Computation (CS1602) · Lecture 11

Complexity, Exceptions and Testing

Instructor: Tao Huang

Part 5: Robustness and the Real World · Fall 2026

| Three Ways a Program Fails 程序的三种失败

Way 方式	Symptom 症状	This lecture 本讲的应对
Too slow	small data is fine, real data never finishes	growth and cost
It crashes	unexpected input ends the program	exception handling
Quietly wrong	no error, wrong answer	assertions and tests

| 三种失败对应三种证据，后面每一段回收一条。



| Two Reports on the Same File 同一份文件的两种处理

Run 运行

Observed 观察

Small sample

both finish instantly

Full file

one finishes, one does not

| 先回答：凭什么说其中一个更好？ What makes one of them better?

Quiz 1 · Counting Iterations



扫码作答 · 1 题

1 数一数内层循环一共执行几次

Submit this round once. 按已教规则推理。

<https://taohuang.info/cs1602/zh/quiz/11/>

| Lecture Structure

- 1 Cost and growth 成本与增长** measure work, not seconds
 - 2 Failure and exceptions 失败与异常** expected problems and clear boundaries
 - 3 Assertions and testing 断言与测试** checks written down as code
 - 4 Five prime methods 素数五法** this week's solo assignment
-



Part V · Robustness and the Real World

输入不会永远合法，数据不会永远干净，规模不会永远小。

What evidence shows that a program can be relied on?



I · Cost and Growth 成本与增长

计时能说明一次运行，说明不了算法之间的差别。

How can two implementations be compared fairly?



Why a Stopwatch Is Not Enough 计时为什么不作为唯一依据

What changes 影响因素

Why it matters 为什么影响结论

Machine speed

the same code takes different seconds on different laptops

Background load

the number moves when the machine is busy

Interpreter version

a newer CPython optimizes different things

Input value

same size, different data, different work

计时只回答「这一次多久」，不回答「这个算法多贵」。



| Count Steps, Not Seconds 数步数

- **Basic operation
- `total = 0` once, for runs `n` times, the body once per pass, return once
- So this function takes about **$n + 2$ steps**, independent of the machine

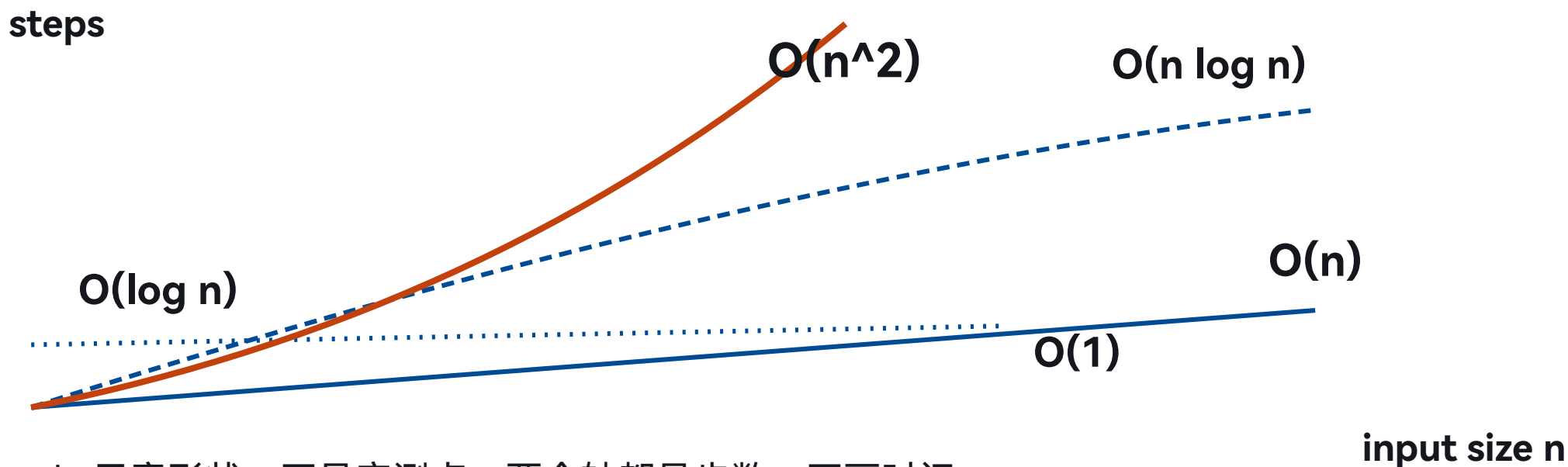


Big-O Keeps the Leading Term 只保留最高阶项

Expression 表达式	Kept 保留	Dropped 丢掉
$n^2 + 3n + 5$	n^2	$3n$ and 5 are lower order
$5n$	n	the constant factor 5
$100n$	n	a bigger constant is still a constant
3	1 (written $O(1)$)	no n at all

| $O(\dots)$ 描述形状；具体多少步要看实现。

| The Ladder of Growth 增长阶



Shapes only 示意形状，不是实测点；两个轴都是步数，不画时间。

$O(1)$ $a[i]$ · $O(\log n)$ binary search · $O(n)$ walking a list · $O(n \log n)$ sorted(a) · $O(n^2)$ a loop inside a loop · $O(2^n)$ naive recursive Fibonacci. 曲线只表示形状，不是实测点。



| A Practical Anchor 一个可用的量级估计

Measurement 测量	n	Best of 3 取最小
Python loop	3,000,000 steps	0.043 s
Built-in sum	2,000,000 items	0.012 s
Steps per second	—	about 7×10^7

| Measured on the build machine 本机构建时实测，同一进程内重复取最小值。

| Prediction · Which Cost Grows Faster?

Which one costs more when n becomes large? 先预测，再往下看。

```
1 def value_at(numbers: list[int], index: int) -> int:
2     total = 0
3     for i in range(len(numbers)):
4         total += 100 * i
5     return total + numbers[index]
```

| Constants Do Not Catch Up 常数追不上阶数

n	100n ($O(n)$)	n^2 ($O(n^2)$)	Smaller 更少的一方
10	1,000	100	$O(n^2)$
100	10,000	10,000	tie
1,000	100,000	1,000,000	$O(n)$
10,000	1,000,000	100,000,000	$O(n)$

| 常数大只输在小规模；阶数高迟早输。

Costs of Python Operations Python 操作的复杂度

Operation 操作	Cost 复杂度	Note 说明
<code>a[i]</code> , <code>len(a)</code>	$O(1)$	position is computed, length is stored
<code>a.append(x)</code> , <code>a.pop()</code>	$O(1)$	touches the end only
<code>a.insert(0, x)</code> , <code>a.pop(0)</code>	$O(n)$	everything after shifts
<code>x in a</code> (list)	$O(n)$	scans from the front
<code>x in s</code> , <code>d[k]</code> (set/dict)	$O(1)$	looked up by hash
<code>sorted(a)</code>	$O(n \log n)$	comparison sort
<code>a[i:j]</code> , <code>s1 + s2</code>	$O(j-i)$, $O(n+m)$	builds a new object

列表查找线性、集合字典常数，是选容器的第一条依据。

List or Set, Measured 实测列表与集合

Container 容器	n = 30,000	n = 60,000
list: x in a	0.0127 s	0.0500 s
set: x in s	0.043 ms	0.043 ms

同一百次查询，只换容器。列表慢约 297 倍，n 翻倍后时间翻约四倍。

| Prediction · Building a String

Is the loop slower, and by how much? 先给出倍数，再往下看。

```
1 words = ["hello"] * 80_000
2
3 result = ""
4 for word in words:
5     result += word
6
7 # or collect and join
8 same = "".join(words)
```

| In-Place Concatenation 就地拼接的实测结果

Kind 写法	n = 20,000	n = 80,000
plain s += w	0.467 ms	0.00166 s
keeping every prefix	0.0235 s	1.1580 s
"".join(words)	0.049 ms	0.195 ms

| CPython 对只有一个引用的字符串原地扩展；中间结果被别处引用时就要复制。

Why Binary Search Is $O(\log n)$ 二分查找为什么是对数

Measurement 测量	n	Best of 3 取最小
1,000 elements	10 comparisons	$\log_2(1000) = 9.97$
1,000,000 elements	20 comparisons	halved 20 times
1,000,000,000 elements	30 comparisons	halved 30 times

Measured on the build machine 本机构建时实测，同一进程内重复取最小值。

I Space Complexity 空间复杂度

- **$O(1)$ space**: a running total — a few variables whatever n is
- **$O(n)$ space**: a list of length n , built before returning
- **The same notation** describes memory
- **Memoisation trades space for time**



I Improve the Order First 先降阶再优化常数

- **1. Change the algorithm
- **2. Change the data structure
- **3. Avoid recomputation
- **4. Use built-ins

Quiz 2 · Logarithmic Growth



扫码作答 · 1 题

1 折半次数与对数的关系

Submit this round once. 按已教规则推理。

<https://taohuang.info/cs1602/zh/quiz/11/>



II · Failure and Exceptions 失败与异常

语法正确不等于运行不会出错。

Which failures belong in the caller's control?

| Syntactically Valid, Still Fails 语法正确仍会失败

code

```
1 x = 10
2 y = 0
3 print(x / y)
```

Actual exception · final line 实际异常

ZeroDivisionError:
division by zero



| Two More Runtime Failures 另外两种运行时失败

Code 代码

Actual exception 实际异常

```
print(value + spam * 3)
```

NameError: name 'spam' is not defined

```
print(text + 2) # text is "2"
```

TypeError: can only concatenate str (not "int") to str

NameError: a name was never defined. **TypeError**: two types cannot be combined. 两种都只有执行到那一行才会发生。

| try and except

code

```
1  def safe_divide(  
2      a: float,  
3      b: float,  
4  ) -> float | None:  
5      try:  
6          return a / b  
7      except ZeroDivisionError:  
8          print("cannot divide by zero")  
9          return None  
10  
11  print(safe_divide(10, 2))  
12  print(safe_divide(10, 0))  
13  print("program continues")
```

output

```
5.0  
cannot divide by zero  
None  
program continues
```

| Catch Specific Types 捕获具体类型

code

```
1 def parse_age(text: str) -> int | None:
2     try:
3         return int(text)
4     except ValueError:
5         print(f"{text!r} is not an integer")
6         return None
7
8 for s in ["18", "abc", "3.14", ""]:
9     print(s, "->", parse_age(s))
```

output

```
18 -> 18
'abc' is not an integer
abc -> None
'3.14' is not an integer
3.14 -> None
'' is not an integer
-> None
```

| Prediction · Handler Order

Which handler runs? 先预测输出顺序。

```
1  def classify(value: str) -> str:
2      try:
3          number = int(value)
4      except Exception:
5          return "general"
6      except ValueError:
7          return "specific"
8      else:
9          return f"ok {number}"
10     finally:
11         print("cleanup")
12
13  print(classify("abc"))
```

| First Match Wins 先匹配先执行

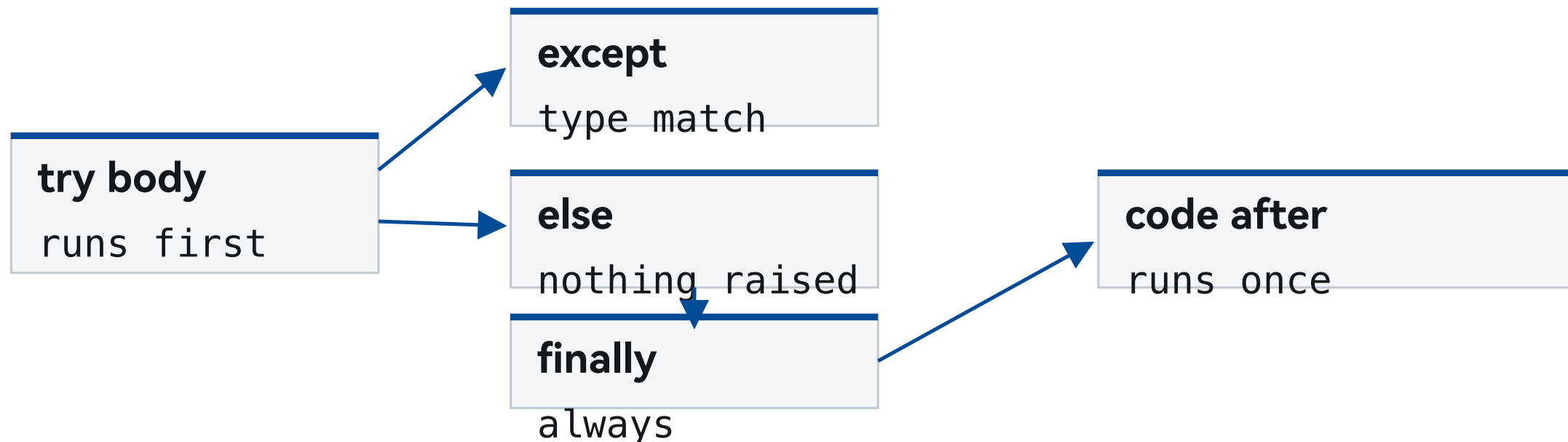
code

```
1 def classify(value: str) -> str:
2     try:
3         number = int(value)
4     except Exception:
5         return "general"
6     except ValueError:
7         return "specific"
8     else:
9         return f"ok {number}"
10    finally:
11        print("cleanup")
12
13 print(classify("abc"))
```

output

```
cleanup
general
```

| else and finally



finally runs on every way out 每条离开路径都执行，return 也一样。

| 实测 10 / 2 走 else 与 finally, 10 / 0 走 except 与 finally: 10 / 2: / ok, result 5.0 / cleanup / 10 / 0: / failed / cleanup

| Prediction · finally and return

What does the function return?

```
1  def f() -> str:
2      try:
3          return "from try"
4      finally:
5          return "from finally"
6
7  print(f())
```

| finally Cannot Be Skipped 清理代码不写 return

code

```
1 def f() -> int:
2     try:
3         return 1 / 0
4     finally:
5         return 42
6
7 print(f())
```

output

42

| raise 主动抛出

code

```
1 def set_age(age: int) -> int:
2     if age < 0:
3         raise ValueError(f"age cannot be negative: {age}")
4     if age > 150:
5         raise ValueError(f"unreasonable age: {age}")
6     return age
7
8 print(set_age(18))
9 print(set_age(-5))
```

Actual exception · final line 实际异常

ValueError:
age cannot be negative: -5

| Custom Exceptions 自定义异常

code

```
1 class InsufficientFunds(Exception):
2     def __init__(self, balance, amount):
3         super().__init__("too low")
4         self.balance = balance
5         self.amount = amount
6
7     def withdraw(balance, amount):
8         if amount > balance:
9             raise InsufficientFunds(balance, amount)
10        return balance - amount
11
12    try:
13        withdraw(100, 500)
14    except InsufficientFunds as e:
15        print("caught:", e)
16        print("short by:", e.amount - e.balance)
```

output

```
caught: too low
short by: 400
```



HALFWAY

Break

Next: assertions, tests, and this week's assignment.

| with and Cleanup 上下文管理与清理

code

```
1 with open("demo.txt", "w") as handle:
2     handle.write("hello")
3 print("closed:", handle.closed)
4
5 try:
6     with open("demo.txt") as reader:
7         raise ValueError("body failed")
8 except ValueError:
9     print("still closed:", reader.closed)
```

output

```
closed: True
still closed: True
```



| Check First or Catch 先检查还是先捕获

Style 风格	Shape 写法	When it reads better 什么时候更清楚
LBYL	if key in d: use it	simple preconditions
EAFP	try it, catch KeyError	the normal path is the main line
Both are correct	choose by intent	neither is a rule

| 看意图，不背规则。

Quiz 3 · Handler Order



扫码作答 · 1 题

1 分支顺序与 finally 的执行位置

Submit this round once. 按已教规则推理。

<https://taohuang.info/cs1602/zh/quiz/11/>



III · Assertions and Testing 断言与测试

程序可以又快又稳，同时结果是错的。

What checks belong inside the program itself?



| assert Writes Down an Assumption 断言写下假设

code

```
1 def average(nums: list[float]) -> float:
2     assert len(nums) > 0, "empty list"
3     return sum(nums) / len(nums)
4
5 print(average([1, 2, 3]))
```

output

2.0

| Prediction · Assertions Under -O

What changes when assertions are disabled?

```
1 def average(nums: list[float]) -> float:
2     assert len(nums) > 0, "empty list"
3     return sum(nums) / len(nums)
4
5 print(average([1, 2, 3]))
```

| Assertions Can Be Removed 断言会被移除

Mode 模式	__debug__	Result 结果
python3 example.py	__debug__ = True	AssertionError: empty list
python3 -O example.py	__debug__ = False	ZeroDivisionError: division by zero

| 断言是开发期的检查，不能校验用户输入；那要用 if 加 raise，不会被优化掉。

Quiz 4 · Assertion Side Effects



扫码作答 · 1 题

1 断言表达式里的副作用

Submit this round once. 按已教规则推理。

<https://taohuang.info/cs1602/zh/quiz/11/>

| Assertions as Documentation 断言作为文档

Place 位置	Written as 写成	Catches 能查出什么
Precondition	<code>assert sorted(nums) == nums</code>	an argument that is out of contract
Postcondition	<code>assert 0 <= i < len(nums) or i == -1</code>	a result the code promised but did not deliver

| 断言描述契约；契约变了，断言跟着变。



| pytest 把测试正规化

- **Files and functions** both start with `test_`
- **Expectations are ordinary assert**
- **`pytest.raises`** when a failure is the expected outcome
- **`python3 -m pip install pytest`**

Running pytest 运行与失败诊断

Actual pytest -q output · real run 真实运行输出 (终端行按面板宽度折行)

```
...F
[100%]
===== FAILURES =====
=====
E      ValueError: cannot divide by zero
mymath.py:6: ValueError: cannot divide by zero
===== short test summary info =====
FAILED test_mymath.py::test_divide_by_zero - ValueError: cannot divide by
zero
1 failed, 3 passed in 0.00s
```

| What to Test 该测什么

Category 类别	Example 例子
Normal	a typical input
Empty	an empty list, an empty string
Single	exactly one element
Boundary	zero, negatives, largest and smallest
Failure	does invalid input raise as agreed

| 每个分支一个用例；空与单个最容易漏。

| Test Code You Did Not Write 给别人的代码写测试

Excerpt · complete definition in notes

```
def longest_word(text):  
    words = text.split()  
    return max(words, key=len)
```

- **word_count("") -> 0**
normal 正常返回
- **longest_word("") -> ValueError** max() got an empty sequence
- **average_word_length("") -> ZeroDivisionError**
division by zero

[Complete runnable example](#) 完整程序

| Prediction · How Many Fail on Empty Input?

How many of the three fail on an empty string?

```
1  def word_count(text: str) -> int:
2      """Count the words."""
3      return len(text.split())
4
5  def longest_word(text: str) -> str:
6      """Return the longest word. On a tie, the first one."""
7      words = text.split()
8      return max(words, key=len)
9
10 def average_word_length(text: str) -> float:
11     """Return the mean word length."""
12     words = text.split()
13     return sum(len(w) for w in words) / len(words)
```

| Expected and Actual 期望与实际

Actual output 实跑输出

```
word_count: 0  
longest_word: ValueError  
average: ZeroDivisionError
```

- **Two of three fail** on the empty string 三个里两个失败
- **The docstrings never mention it**
- **A test writes the expectation down first**

| How the Platform Marks Work 平台如何判分

```
1 def average(nums: list[float]) -> float:  
2     print(sum(nums) / len(nums))  
3  
4 got = average([1, 2, 3])  
5 print(got == 2.0)
```

I Quiz Review · Assertions and Testing

Question	Actual result	Reason
Q4 assert not events.append("hit")	1	append returns None, so the assert passes; the element is already in the list
Q5 lru_cache over a list argument	TypeError: unhashable type: 'list'	a cache must build a key from its arguments

| The Cost of the Five Methods 五种方法的成本

Method 方法	Time 时间复杂度	Space 空间	Idea 思路
brute	$O(n^2)$	$O(1)$	every p tries every d below it
sqrt	$O(n \sqrt{n})$	$O(1)$	a factor below $\sqrt{p}$ is enough
$6k \pm 1$	$O(n \sqrt{n})$	$O(1)$	skip multiples of 2 and 3
sieve	$O(n \log \log n)$	$O(n)$	mark the whole range at once
Miller-Rabin	$O(n \log^3 n)$	$O(1)$	a few modular powers per number

| Measured Prime Counts 素数方法的实测

Method 方法	n = 2000	n = 100000
brute	0.0042 s	too slow
sqrt	0.473 ms	0.0864 s
6k+-1	0.195 ms	0.0288 s
sieve	0.036 ms	0.0021 s
Miller-Rabin	0.455 ms	0.0267 s

| 五种方法在小规模都得 303，大规模都得 9592；先验证再计时。

Quiz 5 · Caching and Keys



扫码作答 · 2 题

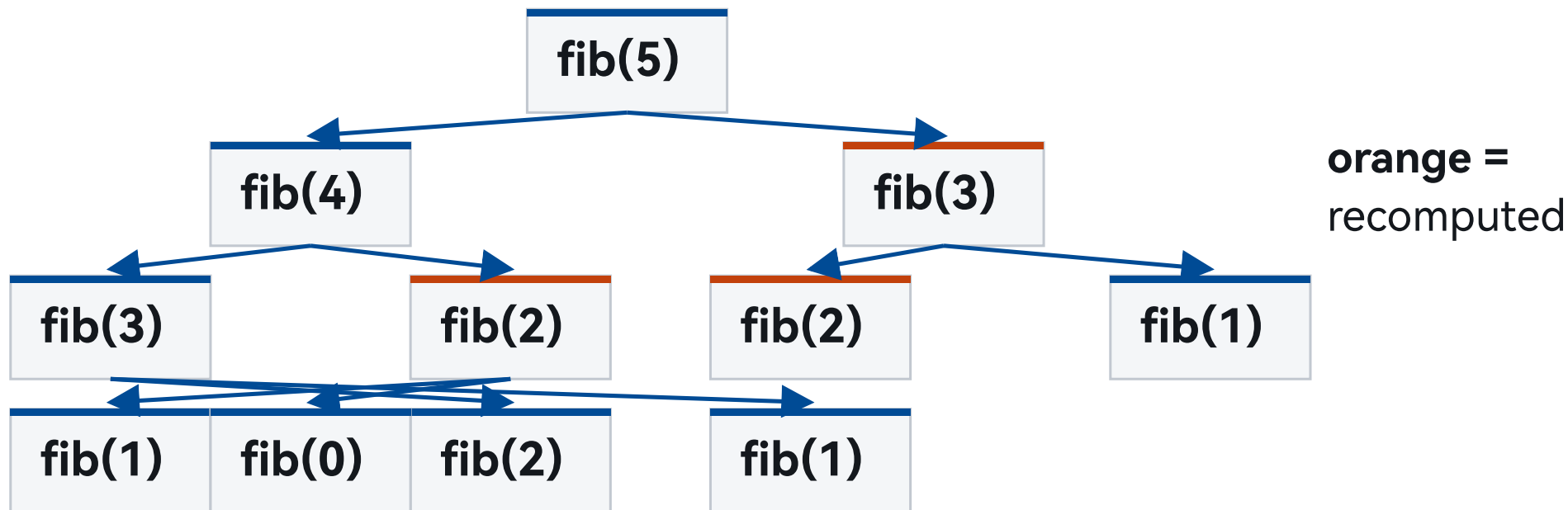
1 缓存要求可哈希的参数

2 缓存键必须唯一确定子问题

Submit this round once. 按已教规则推理。

<https://taohuang.info/cs1602/zh/quiz/11/>

Memoization: Remember Computed Results 记忆化



One call tree 一次调用树; `fib(3)` and `fib(2)` are computed again and again.

实测 `fib(20)` = 6765 calls = 21,891; `fib(25)` = 75025 calls = 242,785; `fib(28)` = 317811 calls = 1,028,457。n 加 3，调用次数大约翻四倍，结果只增长到三倍左右。

The Cache Key Identifies a Subproblem 缓存键与子问题

Key on amount alone. What comes back?

```
1  def ways(amount: int, coins: list[int], index: int) -> int:
2      if amount in memo:
3          return memo[amount]
4      if amount == 0:
5          return 1
6      if amount < 0 or index >= len(coins):
7          return 0
8      memo[amount] = (ways(amount - coins[index], coins, index)
9                      + ways(amount, coins, index + 1))
10     return memo[amount]
11
12 memo: dict[int, int] = {}
13 print("key = amount: ", ways(10, [1, 2, 5, 10], 0))
```

Two Conditions for a Correct Cache 缓存正确的两个条件

Condition 条件

Why 理由

The key identifies the subproblem uniquely

every varying argument belongs in it: (amount, index), not amount

Same input, same output

a function of the clock, of randomness or of outside state cannot be cached

11 = 11, ways(100) = 2156, calls = 801, cache entries = 400; fib(28) = 317811, calls = 55, cache entries = 27

| functools.lru_cache

code

```
1  from functools import lru_cache
2
3  @lru_cache(maxsize=None)
4  def fib(n: int) -> int:
5      if n <= 1:
6          return n
7      return fib(n - 1) + fib(n - 2)
8
9  print(fib(10))
10 i = fib.cache_info()
11 print(i.hits, i.misses, i.currsiz)
12 print(fib(10))
13 print("hits:", fib.cache_info().hits)
14 print(fib(100))
```

output

```
55
8 11 11
55
hits: 9
354224848179261915075
```

| Solo Assignment 2 · Counting Primes

- Five genuine methods
- Range $0 \leq p < n$, $n \leq 100000$; $n \leq 2$
- Verify first, then time
- Miller-Rabin bases 2 and 3 are certain for $n < 1,373,653$
- Three solo assignments: 15% total; this one: 5%

| 本周任务是素数计数；大整数的分组表示留到第12周。



| Choosing a Method 五种方法怎么选

Situation 情形	Method 方法	Why 理由
many numbers, modest range	sieve	one pass marks the whole range
one very large number	Miller-Rabin	$O(1)$ space, a few modular powers
this week, $n \leq 100000$	any trial division	all finish; only the constant differs

| 先看规模和空间，再选方法。



| Lab 11 · Fixed Interfaces 固定接口

Entry 入口

Contract 契约

w11-complexity-measure	comparable timings; no printing
w11-string-build	three builders agree; build_keep is the comparison
w11-memoize	the key holds every varying argument
w11-safe-convert	specific types only; no bare except
w11-config-review	a missing file differs from malformed content
w11-custom-exception	set_celsius raises TemperatureError
w11-tests	four cases per function, one pytest.raises

| Common Misjudgements 常见误判

Written 常见写法

What actually happens 实际发生

`except`: catches everything safely

it swallows typos and `KeyboardInterrupt`

`assert` validates user input

assertions are removed under `-O`

`return` in `finally` just returns early

it replaces the value and discards the exception

`assert (cond, msg)` adds a message

a non-empty tuple is always true

more tests means better tests

covering the branches matters, not the count

read the implementation, then write tests

write the expectation from the docstring first



| Three Kinds of Evidence 小结

- **Growth
- **Boundaries
- **Checks

Reference 参考

完整机制、旧例与提交规格。

After class, not another required lecture.

| Worst, Best and Average Case 最坏最好与平均

Case 情况	Definition 定义	Role here 本课的处理
Worst	the most expensive input at that size	the default
Best	the cheapest input	rarely worth quoting
Average	the expectation over inputs	needs a distribution assumption



| Asymptotic Meaning 渐近含义

- $O(f(n))$ is an upper bound
- $n + 2$ is $O(n)$, and also $O(n^2)$ — we write the tightest bound
- **Dropping constants is safe** only when the input is large and the orders differ
- **Same order means the constant is all that is left**

| Polynomial Evaluation 多项式求值

Actual output 实跑输出

5.4642884629601965e+70

- **Naive 朴素写法** uses x^{**} power every iteration
- **Horner** replaces powers with repeated multiply-and-add
- **Same mathematics**

| List Memory Model 列表内存模型

- Elements are stored contiguously
- Index corresponds to element one to one
- **Together:** $a[i]$ costs $O(1)$; $a.insert(0, x)$ and $a.pop(0)$ cost $O(n)$
- **Measured** inserting at the front: $n = 5,000$ in 0.0015 s, 10,000 in 0.0046 s, 20,000 in 0.0199 s



| Hash Table 哈希表

Idea 想法

Consequence 推论

A position function computes where a value is stored

lookup does not scan

More slots than elements

space for time

Two keys can land on one slot

collisions exist and have standard remedies

| hash() and Contracts 哈希契约

Expression 表达式

Result 实测结果

`hash(1) == hash(1.0)`

True

`hash(True) == hash(1)`

True

`hash((1, 2))`

works

`hash([1]) / hash({"a": 1})`

TypeError: unhashable type

a class that defines `__eq__` only

`__hash__` becomes None

| 相等则同哈希；可变对象不做键。



| Custom `__hash__` 自定义哈希

- Equal objects must hash equally
- The hash need not be unique
- Mutable objects make unsafe keys
- Easier route

| Binary Search 二分查找

Actual output 实跑输出

2 -1

- **The +1 and -1 keep it moving**
边界加减一，否则死循环
- **Sorted input is required**
- **-1 means absent**



| Recursion Cost 递归成本

n	Calls 调用次数	Answer 结果
20	21,891	6,765
25	242,785	75,025
28	1,028,457	317,811

| Memoized Fibonacci 记忆化斐波那契

- Dictionary cache
- List cache
- Both are $O(n)$ time and $O(n)$ space
- `@lru_cache(maxsize=...)` caps the entries



| Space and Recursion 空间与递归

- Each recursive call saves its own state
- Naive fib occupies $O(n)$ of stack
- An iterative version finishes in $O(1)$
- Generators take this further

| raise Forms 抛出形式

Actual output · real run 实跑输出

```
value: alpha  
logged: 'missing'  
the caller decides
```

- **Class plus argument**
- **Bare raise**
- **from keeps the cause**
- **Returning None instead**

| 裸 raise 记下信息后继续往上抛；异常链用 from 保留因果，讲义有完整例子。

| Multiple Handlers 多分支

Form 写法

Meaning 含义

`except (KeyError,
IndexError):`

tuple form

`except:`

catches everything, including KeyboardInterrupt

`except Exception as error:`

catch-all with a message

sibling clauses

do not catch each other



| Custom Exception Hierarchy 自定义异常层级

Level 层级	Purpose 作用
<code>class ConfigError(Exception)</code>	a base for one family of problems
<code>class ConfigMissing(ConfigError)</code>	callers can catch the whole family or one subclass
fields	carry the values the handler needs, such as the rejected value

| Resource and OS Errors 资源与系统错误

Exception 异常

Typical source 常见来源

`OSError`, `FileNotFoundError`,
`PermissionError`

files, directories, permissions

`ValueError`

a failed conversion, such as `int("abc")`

`TypeError`

mismatched types

`KeyError` / `IndexError`

missing key / index out of range

| assert Forms 断言写法

Form 写法

Result 结果

`assert condition`

raises `AssertionError` with no message

`assert condition, "message"`

the message is shown

`assert (condition, "message")`

always true

side effects

still happen

| 消息写在逗号后面，不要包进括号。

I `__debug__` and Optimized Mode 优化模式

Setting 设置

Effect 作用

`__debug__`

a constant; False under optimisation

`python3 -O`

assert statements are removed

`python3 -OO`

also drops docstrings

`PYTHONOPTIMIZE`

the same effect through the environment

cached bytecode

the file name carries the optimisation level

| Assertion Pitfalls 断言的误用

Misuse 误用

What goes wrong 后果

validating user input

removed under `-O`, so the check disappears

handling an expected failure

the bug is treated as a user error

side effects in the expression

the effect happens even when the assert passes

catching `AssertionError`

the bug is hidden

| pytest Conventions and Options 约定与选项

Item 项目	Meaning 含义
<code>-v</code>	show every test name and result
<code>-q</code>	progress dots and a summary only
<code>-k expression</code>	run only tests whose names match
<code>pytest.raises(Error)</code>	expect the given exception
<code>pytest.raises(Error, match="...")</code>	also check the message
<code>pytest.approx(x) == y</code>	float comparison without precision surprises



| Notes and a Real Suite 讲义示例与真实测试

- The notes run as single files
- pytest needs discovery
- So the notes show a checking loop
- The real pytest output is quoted verbatim

| Platform Grading 平台判分

Situation 情形

Result 结果

the function returns a value

compared against the expected value

the function prints instead

returns None; `None == expected` is never true

the expected outcome is an exception

the test expects the type, not a return value

| Edge Case Checklist 边界清单

Category 类别	Examples 例子
normal	a typical input
empty	an empty list, an empty string
single	exactly one element
boundary	0, negatives, largest and smallest, equal bounds
repetition	duplicate elements, consecutive spaces
failure	does invalid input raise as agreed

| textstats Under Test 测试第三方模块

Actual output 实跑输出

2

ValueError

- **Repeated spaces do not make an empty word** 连续空格不算单词
- **The docstring never pins down the empty string**
- **The discrepancy is the finding**



| Prime Counting · Brute and Sqrt 暴力与平方根剪枝

- **Brute**
- **for...else** runs the else when the loop was never broken
- **Sqrt**
- **Measured** $n = 2000$: brute 0.0042 s, sqrt 0.00047 s



| Prime Counting · $6k \pm 1$ 剪枝

- Rule out multiples of 2 and 3 first
- What remains is $6k \pm 1$
- Candidates become 5, 7, 11, 13, ...
- Order unchanged at $O(n \sqrt{n})$, about three times faster



| Prime Counting · Sieve 筛法

- Mark the whole range at once
- Start marking at $p * p$
- $O(n \log \log n)$ time, $O(n)$ space
- **Measured** $n = 100000$ in 0.00212 s, the fastest of the five

| Prime Counting · Miller–Rabin

- Write $n-1$ as $d \times 2^r$
- Test $\text{pow}(a, d, n)$ for a few bases
- $\text{pow}(\text{base}, d, n)$ is the three-argument built-in
- Bases 2 and 3 are certain for $n < 1,373,653$
- $O(\log^3 n)$ per number, $O(1)$ space

| Timing Fairly 公平计时

Requirement 要求

Reason 理由

verify correctness first

all five must agree at the same small size

fix the size and the machine

otherwise the numbers are not comparable

warm up, then take the minimum

avoid scheduler noise

a slow method may use a smaller size but it may not swap in another algorithm



| Solo Assignment 2 · Specification 提交规格

Item 项目	Agreement 约定
five functions	count_primes_brute / _sqrt / _6k / _sieve / _mr
range	$0 \leq p < n$; $0 \leq n \leq 100000$
small values	$n \leq 2$ returns 0
invalid input	a negative n raises ValueError
report	method notes, timing table, call counts, usage declaration
weight	three solo assignments: 15% total; this one 5%

| Lab 11 · Required Boundaries 必须满足的边界

Entry 入口

Required boundary 边界

w11-complexity-measure	the helper returns the smallest of repeats
w11-string-build	the kept-prefix version is the comparison
w11-memoize	count_ways and count_ways_memo agree
w11-safe-convert	unexpected failures must escape
w11-config-review	the file is closed on every path
w11-custom-exception	a rejected update keeps the old state
w11-tests	the suite must fail a wrong module



| Summary and Connection 小结与衔接

Thread 主线

Answer 回答

Too slow

count steps and keep the leading term

It crashes

catch what you can handle, raise what you cannot

Quietly wrong

assertions inside, tests outside

This week

five prime methods, with correctness before timing

Next

randomness, Unicode and files — messier input