A Simplified Git

ACM班2015数据结构大作业

陈乐群 柯嵩宇

写程序?

普通青年

```
int find(int *a, int n, int value) {
  for (int i = 0; i < n; ++i)
    if (i == value)
      return i;
    return -1;
}
```

普通青年

```
int find(int *a, int n, int value) {
  for (int i = 0; i < n; ++i)
    if (a[i] == value)
      return i;
    return -1;
}
```

普通青年

```
int find(int *a, int n, int value) {
    int lef = 0, rig = n;
    while (rig - lef > 0) {
        const int mid = (lef+rig)/2;
        if (a[mid] < lef) mid = lef;
        else mid = rig;
    }
}
```

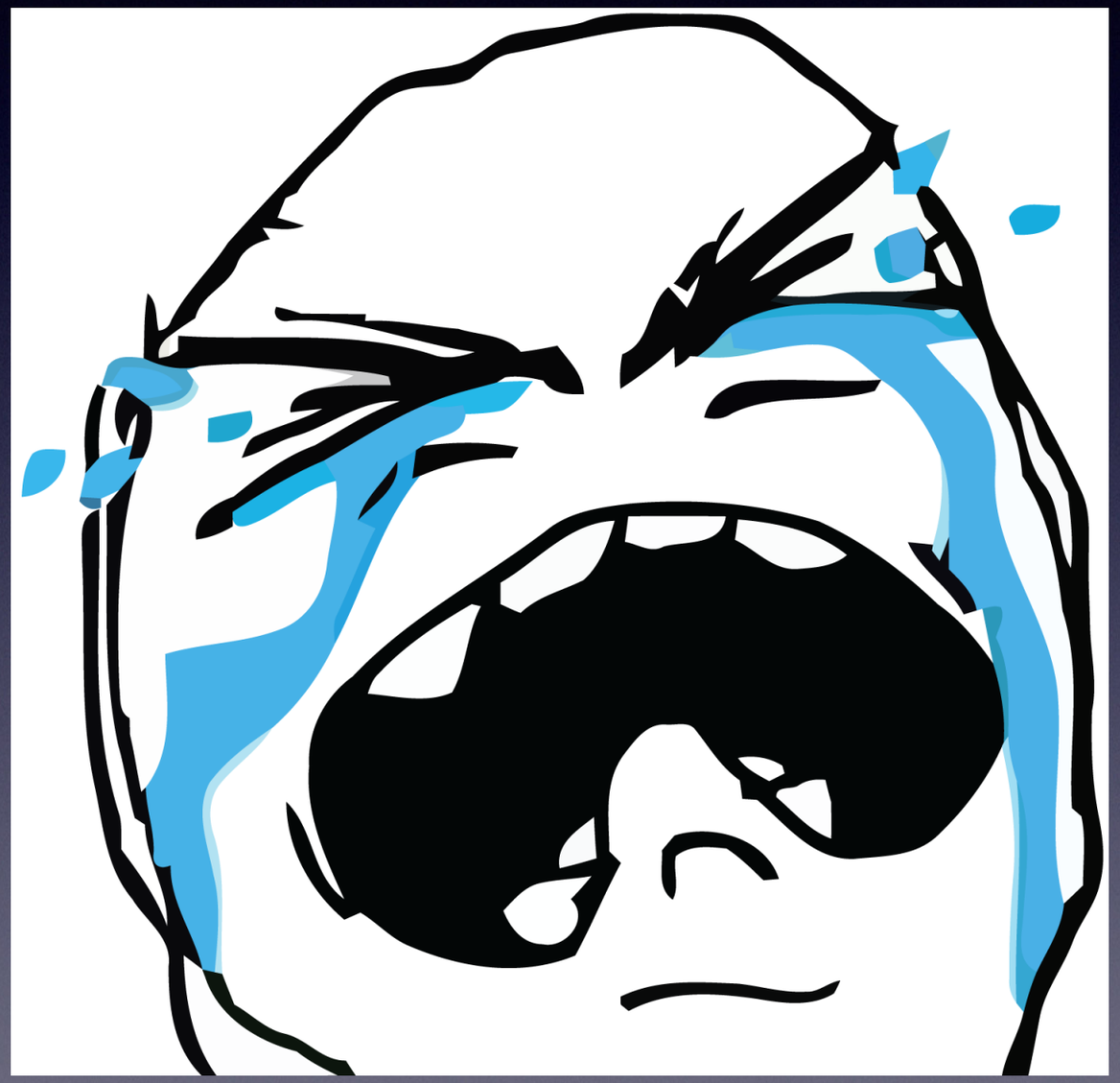
普通青年

```
int find(int *a, int n, int value) {
    int lef = 0, rig = n;
    while (rig - lef > 0) {
        const int mid = (lef+rig)/2;
        if (a[mid] < value) mid = lef;
        else mid = rig;
    }
}
```

普通青年

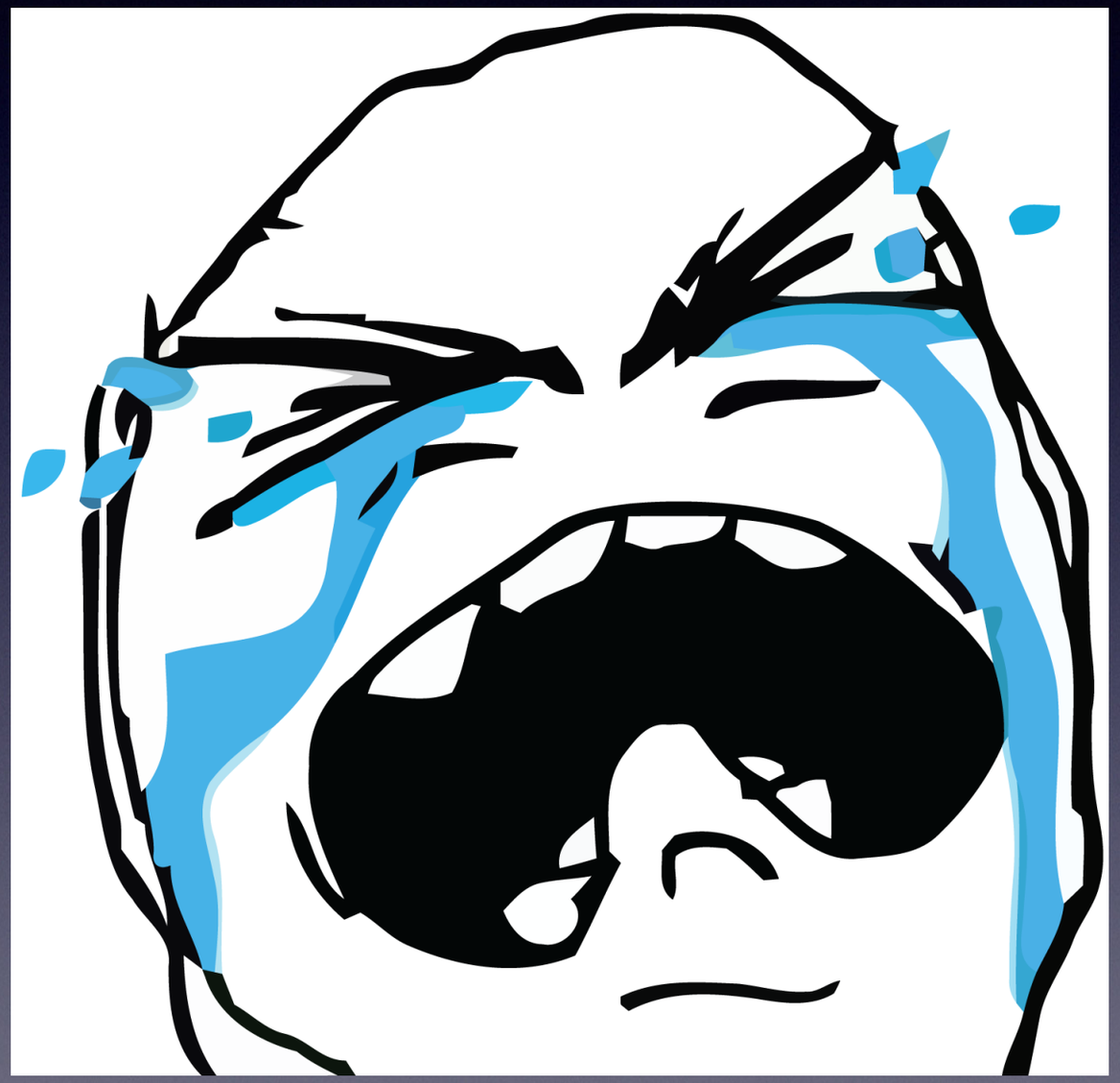
```
int find(int *a, int n, int value) {
    int lef = 0, rig = n;
    while (rig - lef > 0) {
        const int mid = (lef+rig)/2;
        if (a[mid] < value) lef = mid;
        else rig = mid;
    }
}
```

普通青年



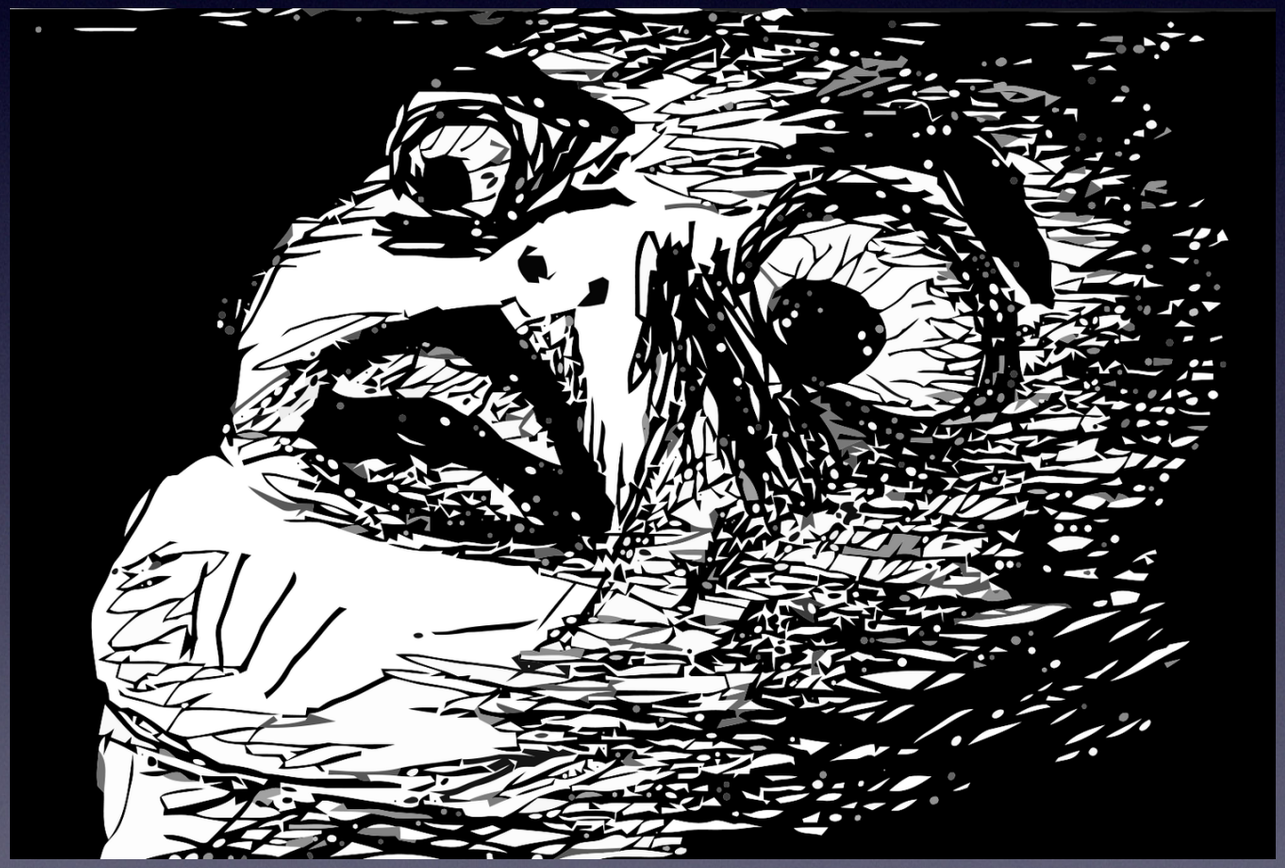
普通青年

之前那个可以用的呢?



普通青年

之前那个可以用的呢?



文艺青年

文艺青年

- ArrayList.h

文艺青年

- ArrayList.h
- ArrayList2.h

文艺青年

- ArrayList.h
- ArrayList2.h
- ArrayList3.h

文艺青年

- ArrayList.h
- ArrayList2.h
- ArrayList3.h
- ArrayList= =.h

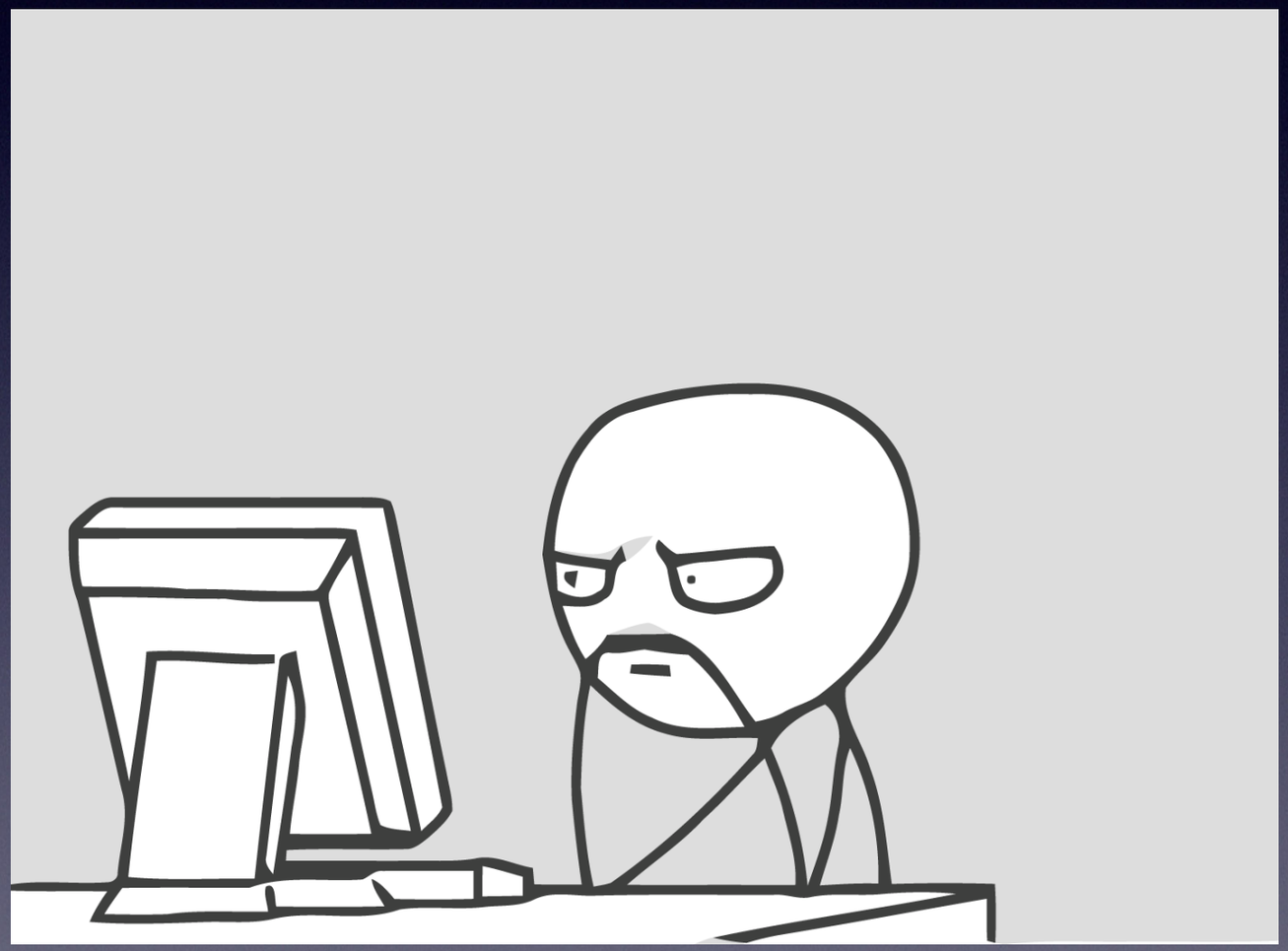
文艺青年

- ArrayList.h
- ArrayList2.h
- ArrayList3.h
- ArrayList= =.h
- ArrayList=_=.h

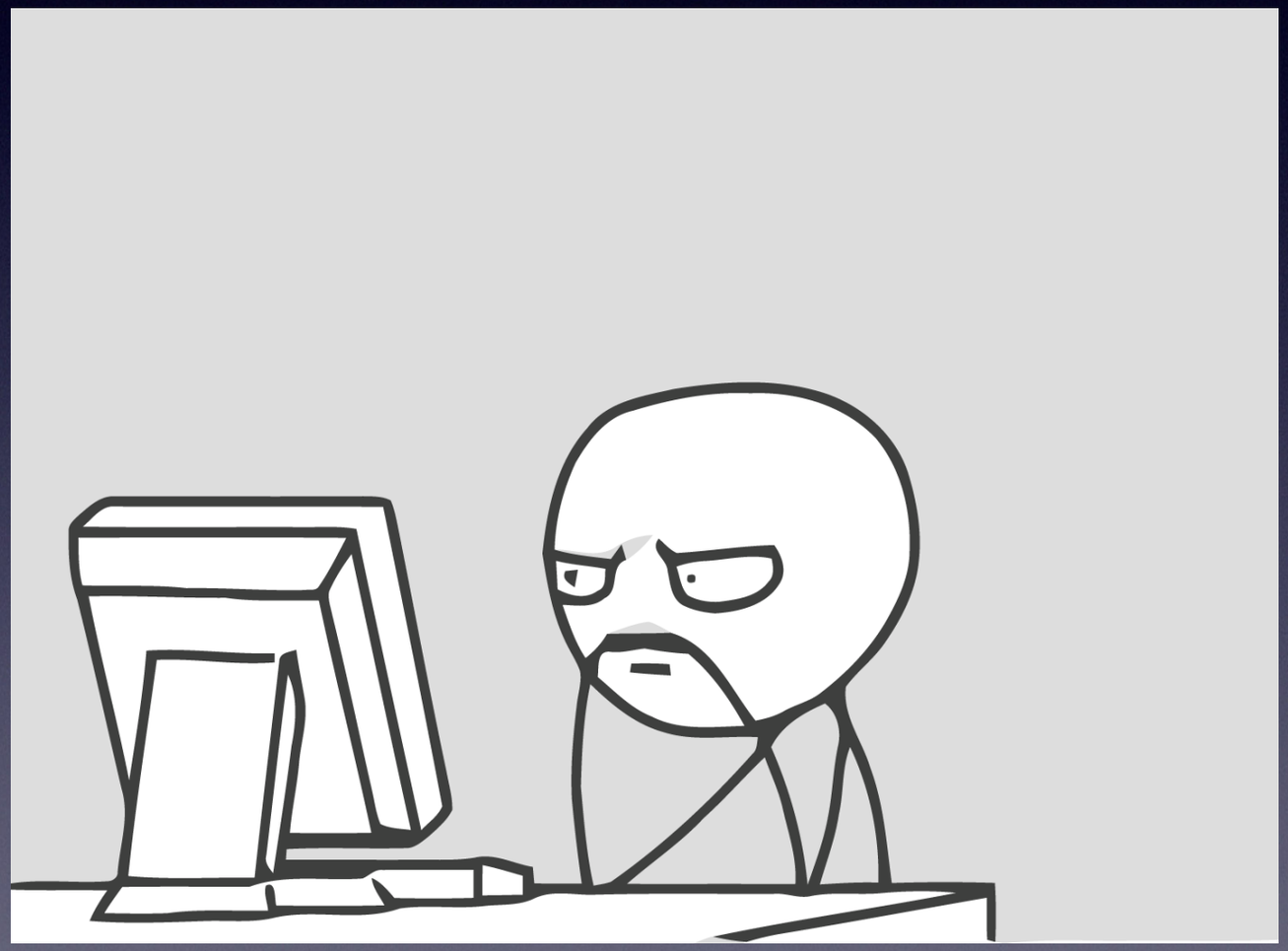
文艺青年

- ArrayList.h
- ArrayList2.h
- ArrayList3.h
- ArrayList= =.h
- ArrayList=_=.h
- ArrayListT_T.h

文艺青年

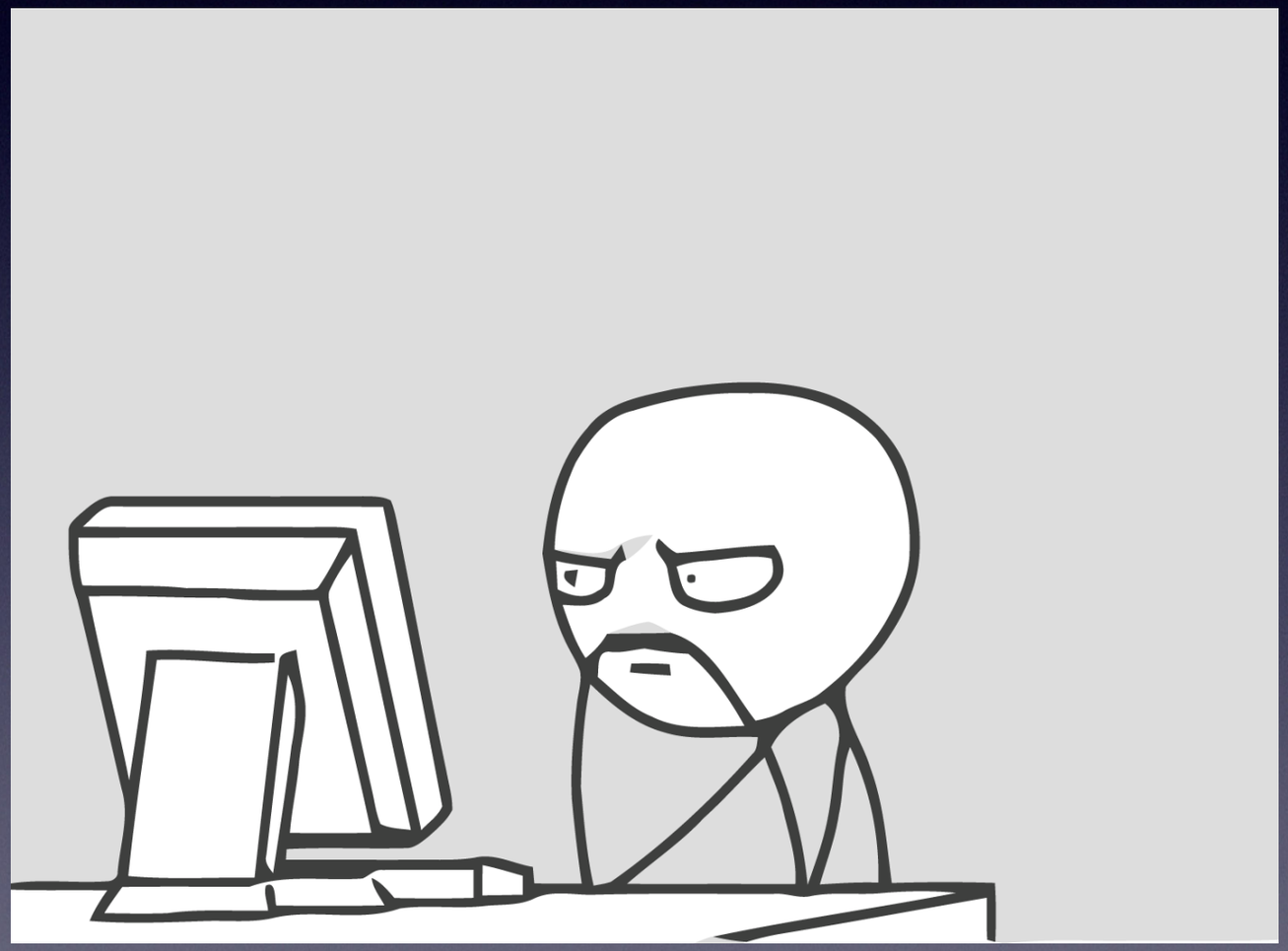


文艺青年



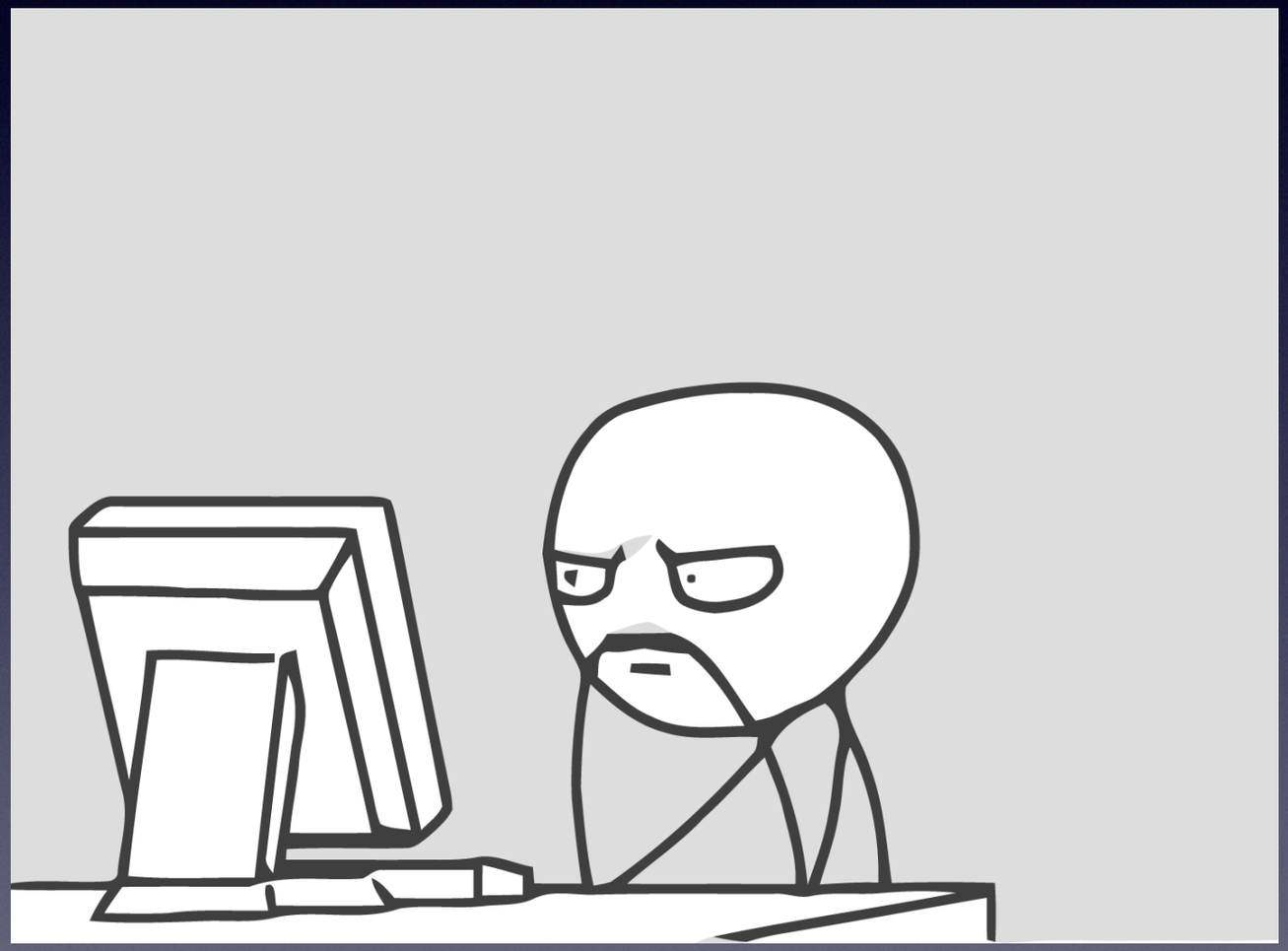
- 好乱

文艺青年



- 好乱
- 好麻烦

文艺青年



- 好乱
- 好麻烦
- 我改了什么?

二逼青年

未命名.cpp

```
1. int main() {
2.     scanf("%d%d", &n, &m);
3.     printf("%d\n", n+m);
4. }
```

二逼青年

未命名.cpp

二逼青年

未命名.cpp

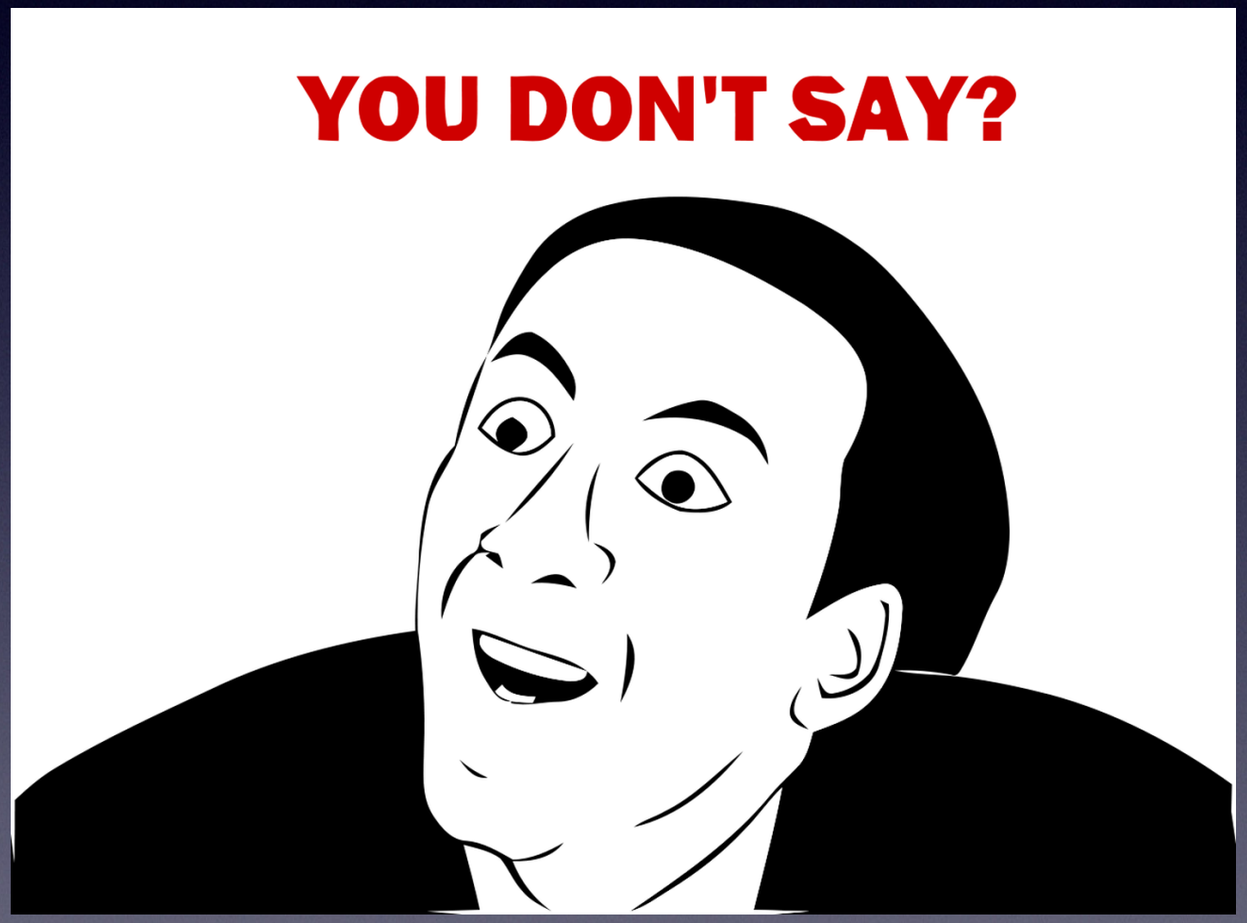
```
1. int main() {
2.     scanf(“%d”, &n);
3.     int sum = 0;
4.     for (int i = 0, x; i < n; ++i) {
5.         scanf(“%d”, &x);
6.         sum += x;
7.     }
8.     printf(“%d”, sum);
9. }
```

二逼青年

未命名.cpp

二逼青年

对于一个工程怎么办?



SIT 能做什么?

版本控制系统

版本控制系统

- 保存每个文件的各个版本

版本控制系统

- 保存每个文件的各个版本
- 方便地查看各个版本的差异

版本控制系统

- 保存每个文件的各个版本
- 方便地查看各个版本的差异
- 轻松地检出任意版本及其中任意的文件

版本控制系统

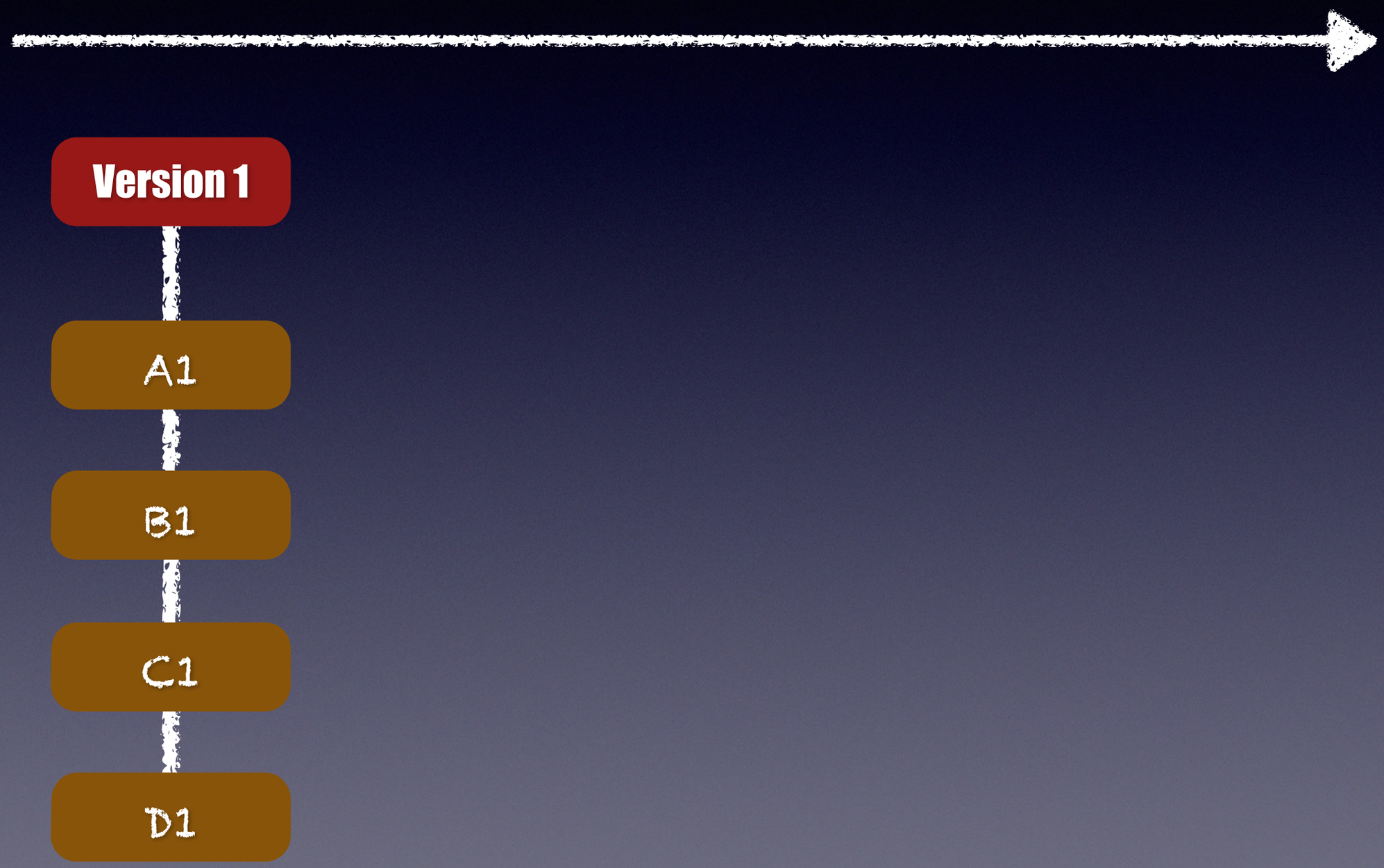
- 保存每个文件的各个版本
- 方便地查看各个版本的差异
- 轻松地检出任意版本及其中任意的文件
- 简单

版本控制系统

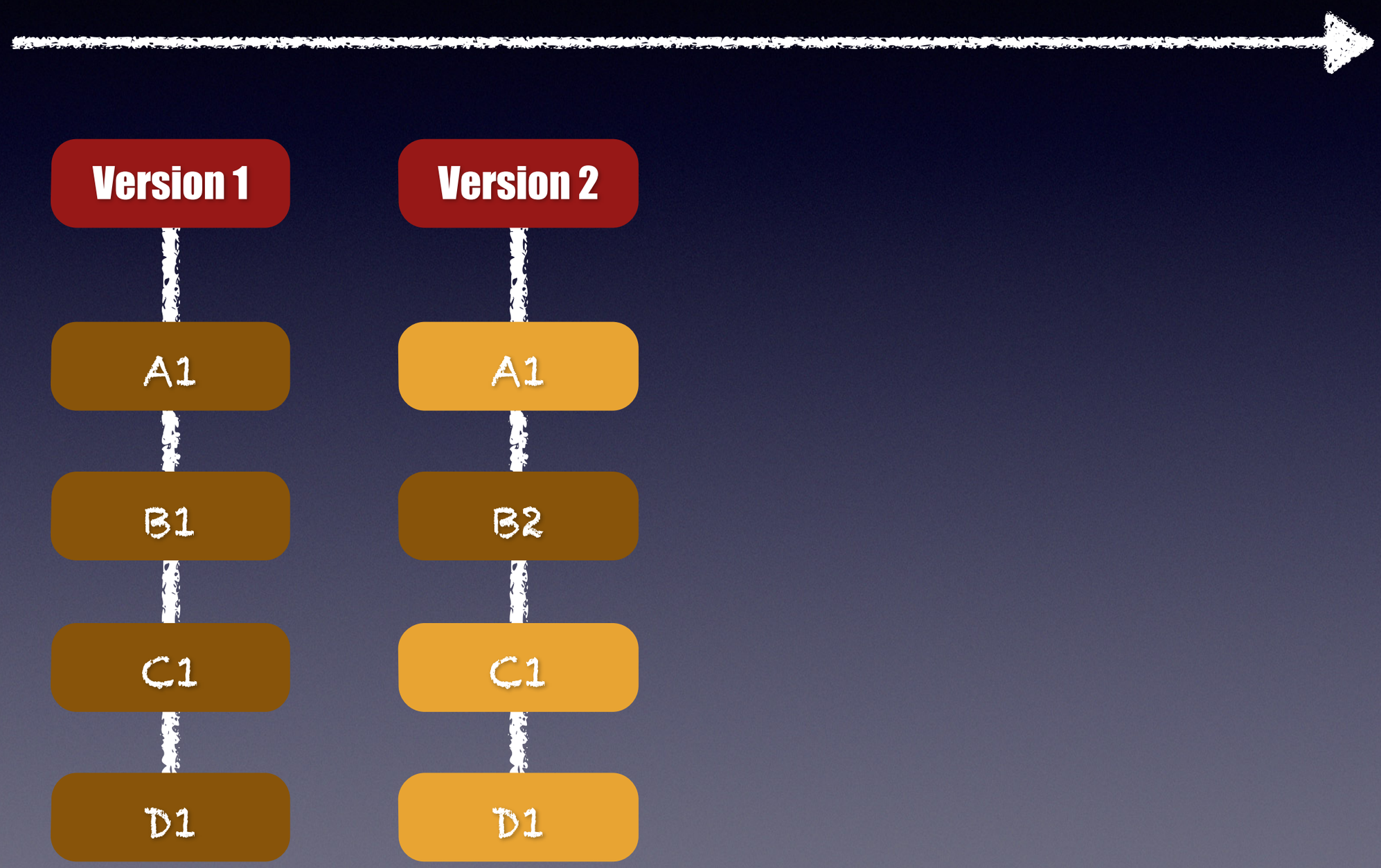
- 保存每个文件的各个版本
- 方便地查看各个版本的差异
- 轻松地检出任意版本及其中任意的文件
- 简单
- 占用空间少

版本控制系统

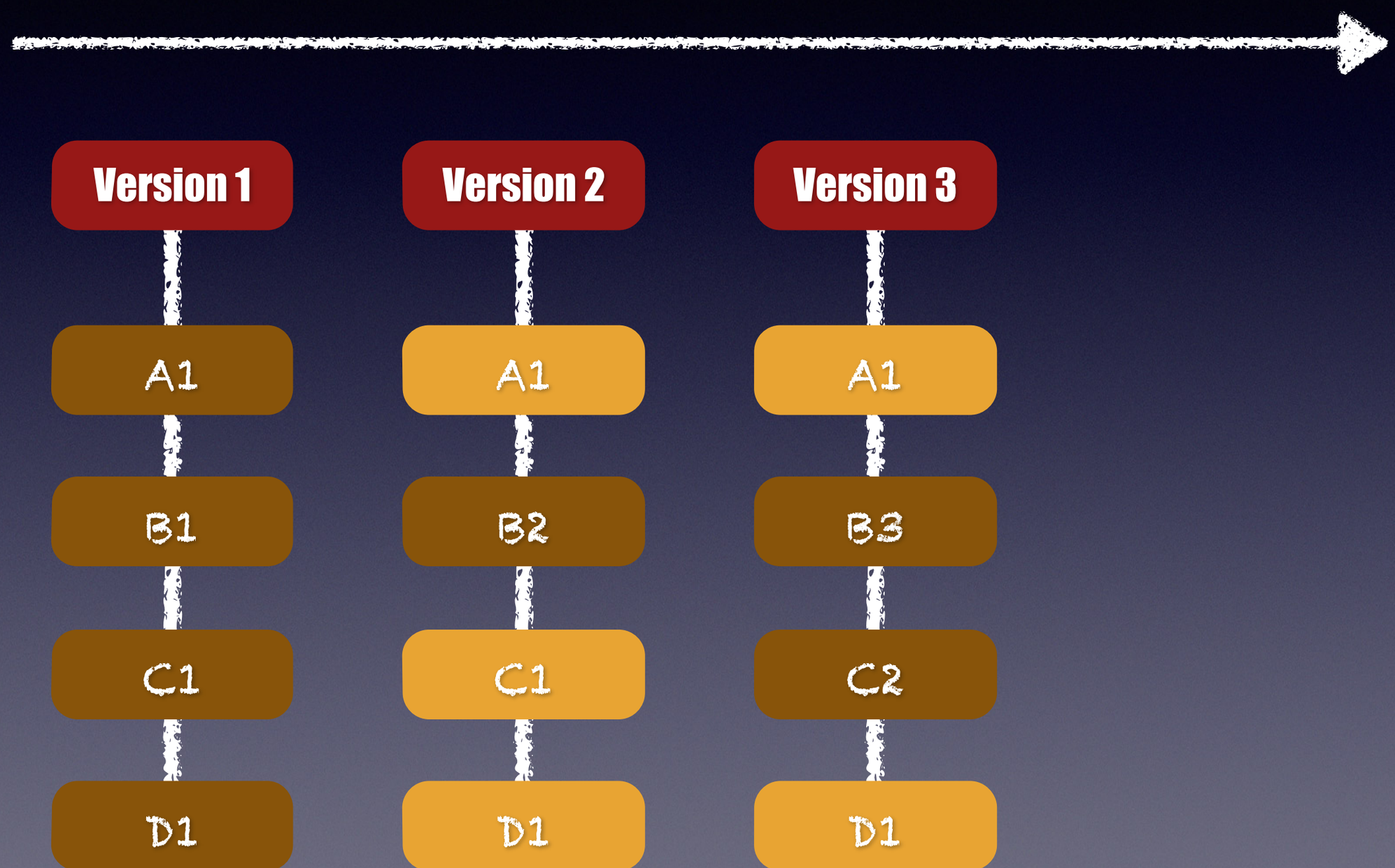
版本控制系统



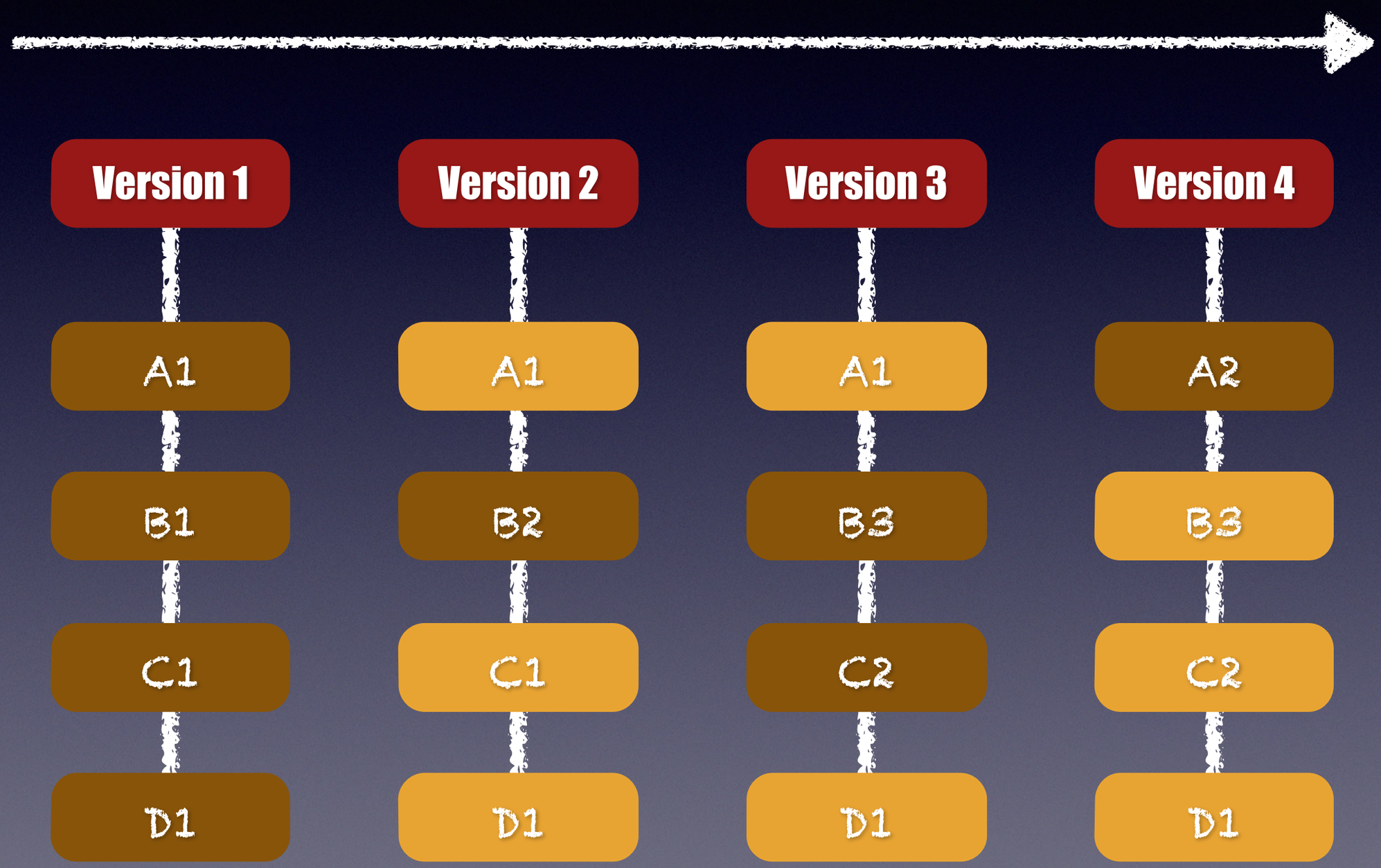
版本控制系统



版本控制系统



版本控制系统


Version 1
Version 2
Version 3
Version 4
A1
B1
C1
D1
A1
B2
C1
D1
A1
B3
C2
D1
A2
B3
C2
D1

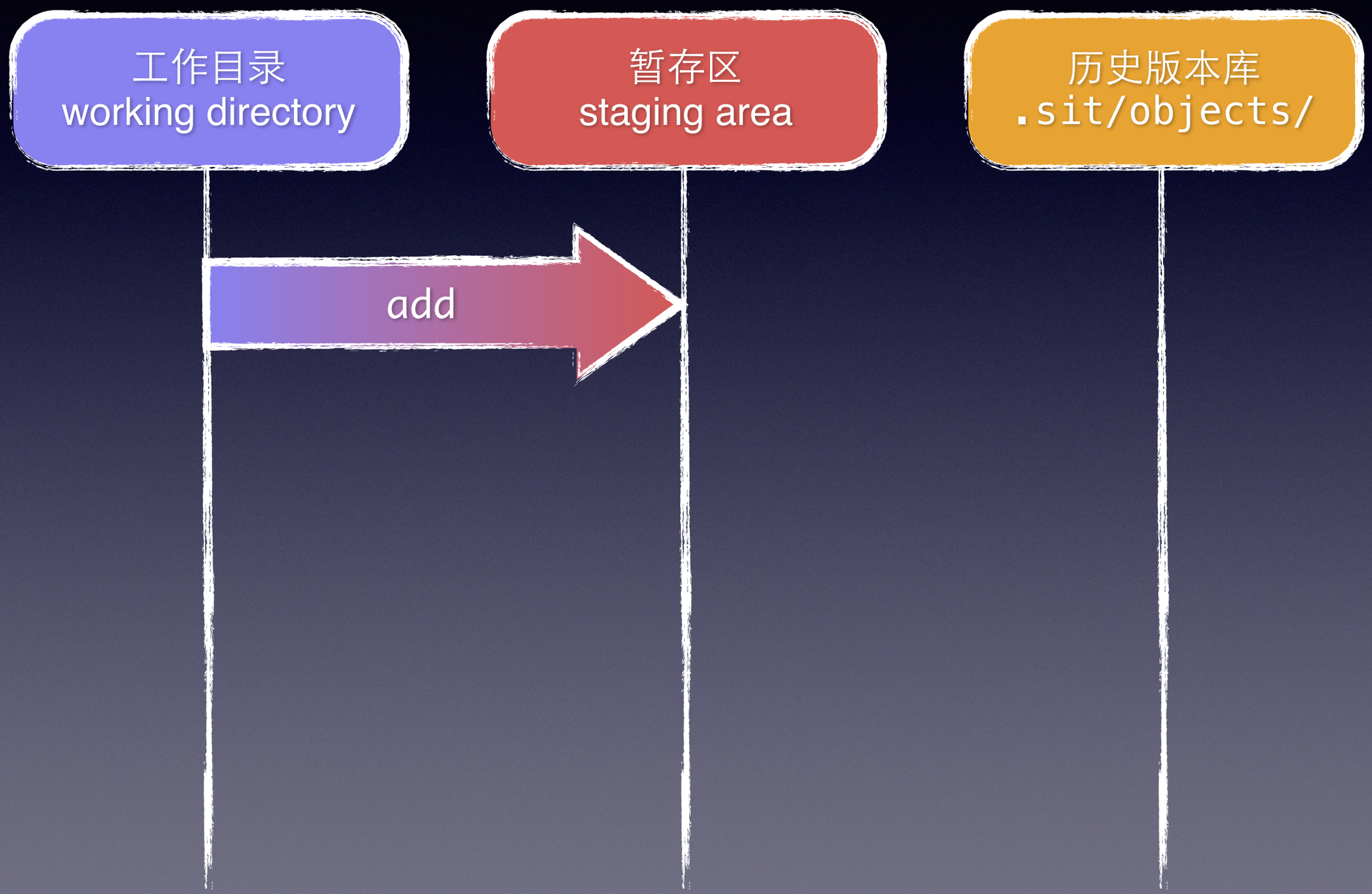
版本控制系统

工作目录
working directory

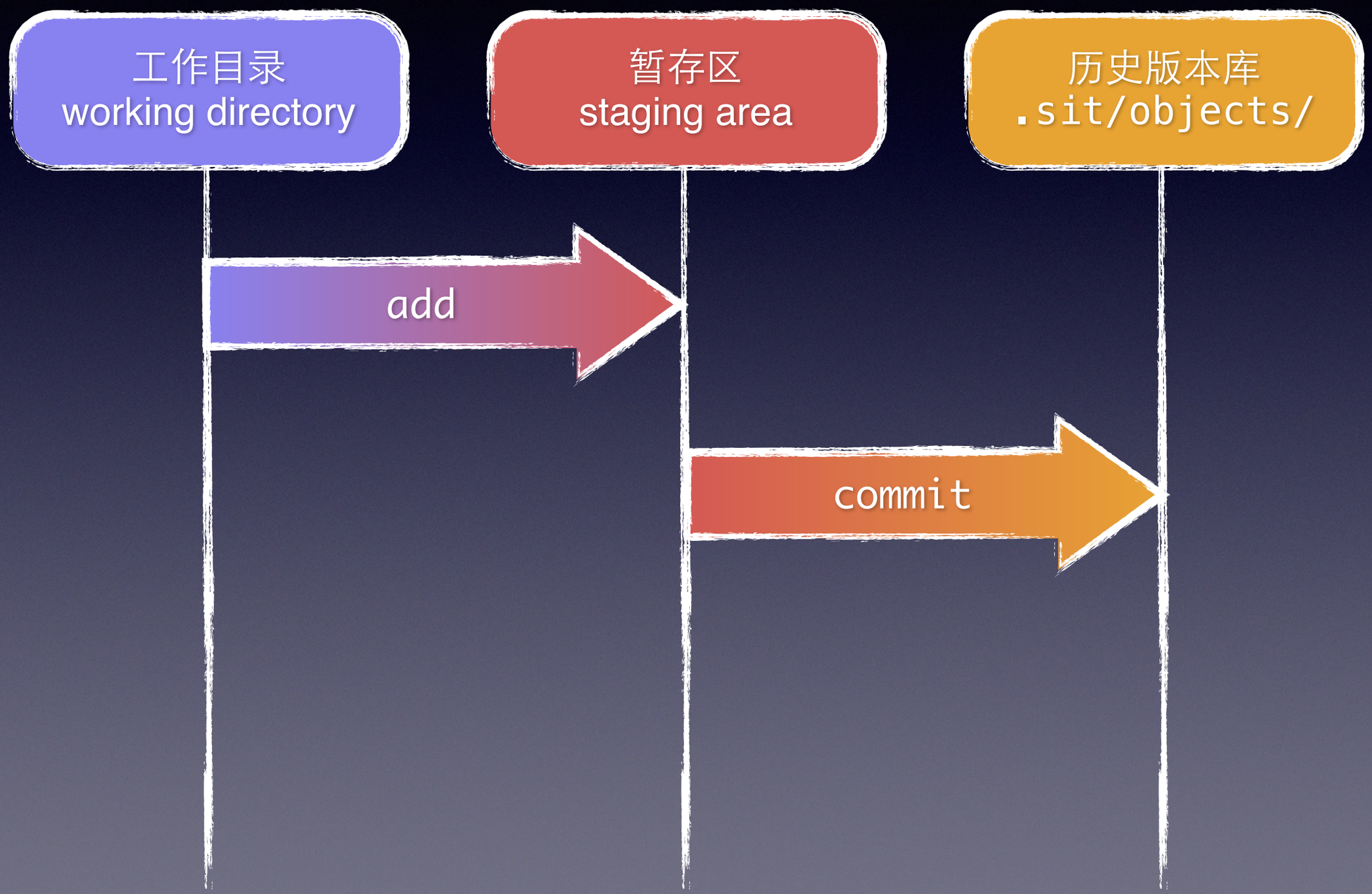
暂存区
staging area

历史版本库
`.sit/objects/`

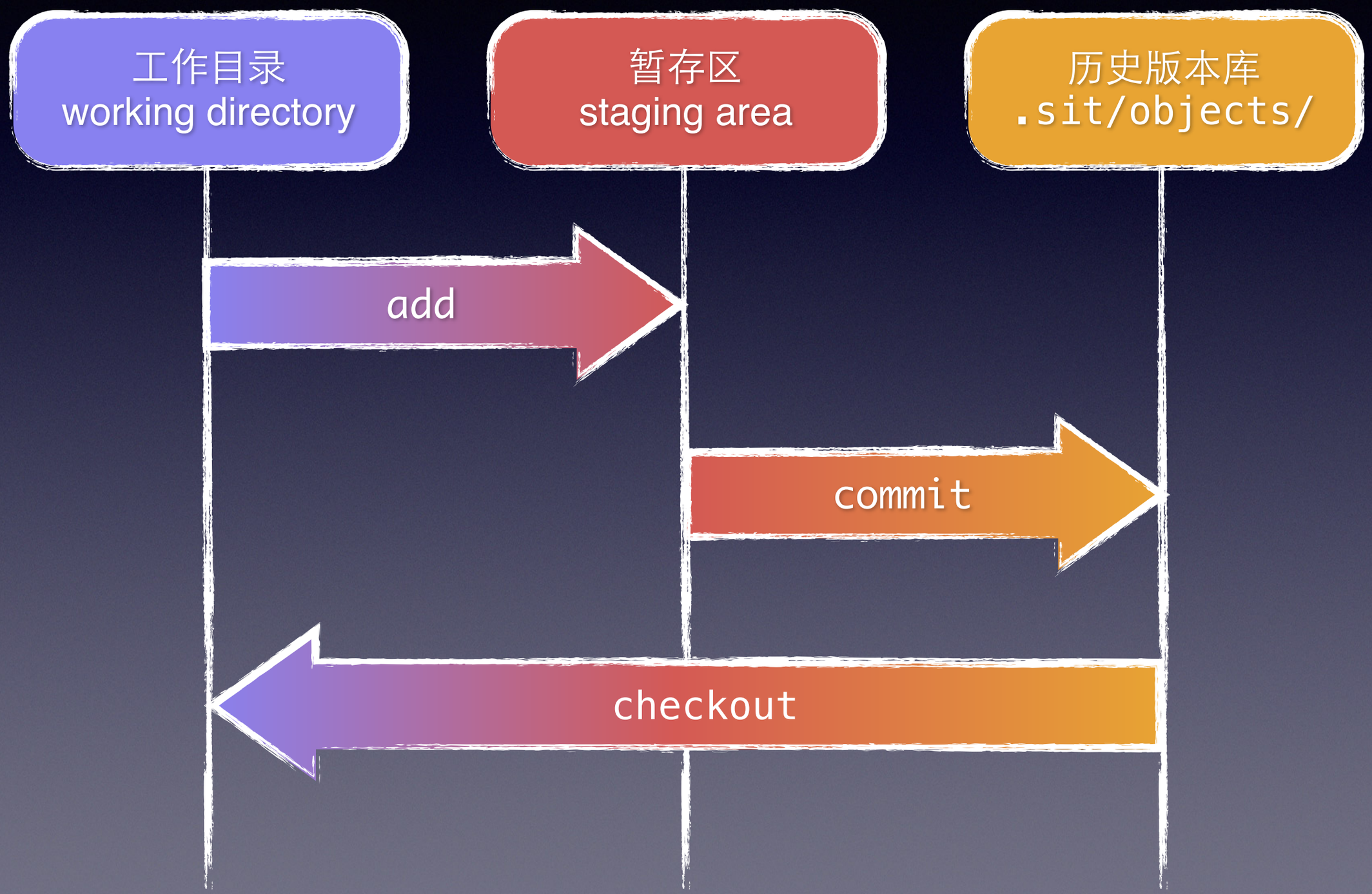
版本控制系统


工作目录
working directory
暂存区
staging area
历史版本库
.sit/objects/
add

版本控制系统


工作目录
working directory
暂存区
staging area
历史版本库
.sit/objects/
add
commit

版本控制系统


工作目录
working directory
暂存区
staging area
历史版本库
.sit/objects/
add
commit
checkout

演示?

SIT的实现?

SIT objects

SIT objects

SIT objects

commit	size
tree	c4ec5
parent	a149b
author	Lequn Chen
fixed a bug. i think my array list runs faster than Hong Kong journalist.	

SIT objects

commit	size
tree	994eb
parent	a8d76
author	Lequn Chen
use copy-and-swap idiom	

commit	size
tree	c4ec5
parent	a149b
author	Lequn Chen
fixed a bug. i think my array list runs faster than Hong Kong journalist.	

SIT objects

commit	size
tree	3c51d
parent	27cfb
author	Lequn Chen
new copy constructor	

commit	size
tree	994eb
parent	a8d76
author	Lequn Chen
use copy-and-swap idiom	

commit	size
tree	c4ec5
parent	a149b
author	Lequn Chen
fixed a bug. i think my array list runs faster than Hong Kong journalist.	

SIT objects

commit
size
tree
3c51d
parent
27cfb
author
Lequn Chen
new copy constructor
commit
size
tree
994eb
parent
a8d76
author
Lequn Chen
use copy-and-swap idiom
commit
size
tree
c4ec5
parent
a149b
author
Lequn Chen
fixed a bug. i think my array list runs faster than Hong Kong journalist.
tree
size
blob 5b1d3 ArrayList.h
blob cdc8b ArrayList.cpp
tree 03e78 lib
tree 911e7 test
blob cba0a README

SIT objects

SIT objects

commit	size
tree	c4ec5
parent	a149b
author	Lequn Chen
fixed a bug. i think my array list runs faster than Hong Kong journalist.	

tree		size
blob	5b1d3	ArrayList.h
blob	cdc8b	ArrayList.cpp
tree	03e78	lib
tree	911e7	test
blob	cba0a	README

SIT objects

commit	size
tree	c4ec5
parent	a149b
author	Lequn Chen
fixed a bug. i think my array list runs faster than Hong Kong journalist.	

tree		size
blob	5b1d3	ArrayList.h
blob	cdc8b	ArrayList.cpp
tree	03e78	lib
tree	911e7	test
blob	cba0a	README

tree		size
tree	1de87	test_list.cpp
tree	3f724	test_map.cpp
tree	d218c	test_set.cpp
tree	c83d7	test.h
tree	27c61	test.cpp

tree		size
tree	3d7ee	jemalloc
tree	1ff83	murmurhash
tree	083af	libz
tree	b87ea	cjson
tree	de46d	crypto

SIT objects

SIT objects

commit	size
tree	c4ec5
parent	a149b
author	Lequn Chen
fixed a bug. i think my array list runs faster than Hong Kong journalist.	

tree		size
blob	5b1d3	ArrayList.h
blob	cdc8b	ArrayList.cpp
tree	03e78	lib
tree	911e7	test
blob	cba0a	README

SIT objects

commit	size
tree	c4ec5
parent	a149b
author	Lequn Chen
fixed a bug. i think my array list runs faster than Hong Kong journalist.	

tree		size
blob	5b1d3	ArrayList.h
blob	cdc8b	ArrayList.cpp
tree	03e78	lib
tree	911e7	test
blob	cba0a	README

SIT objects

commit	size
tree	c4ec5
parent	a149b
author	Lequn Chen
fixed a bug. i think my array list runs faster than Hong Kong journalist.	

tree	size	
blob	5b1d3	ArrayList.h
blob	cdc8b	ArrayList.cpp
tree	03e78	lib
tree	911e7	test
blob	cba0a	README

tree	size	
tree	1de87	test_list.cpp
tree	3f724	test_map.cpp
tree	d218c	test_set.cpp
tree	c83d7	test.h
tree	27c61	test.cpp

blob | **size**

```
#include “../ArrayList.h”
#include <iostream>
#include <string>
using std::string;

int main() {
    ArrayList<string> v;
    v.push_back(“test”);
    v.push_back(“toast”);
}
```

blob | **size**

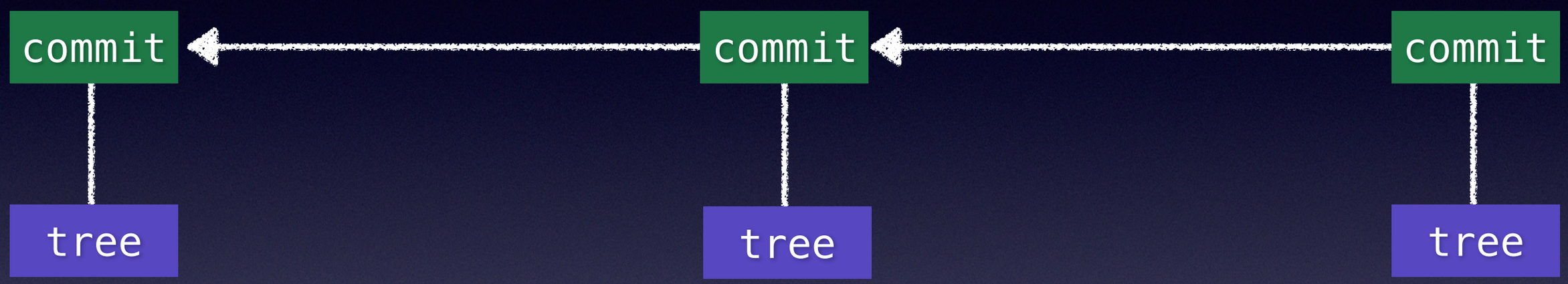
```
#ifndef ARRAYLIST_H
#define ARRAYLIST_H

#include <cstdlib>
#include <string>
#include <vector>
#include <iterator>
#include <algorithm>
#include “ElementNotFound.h”
```

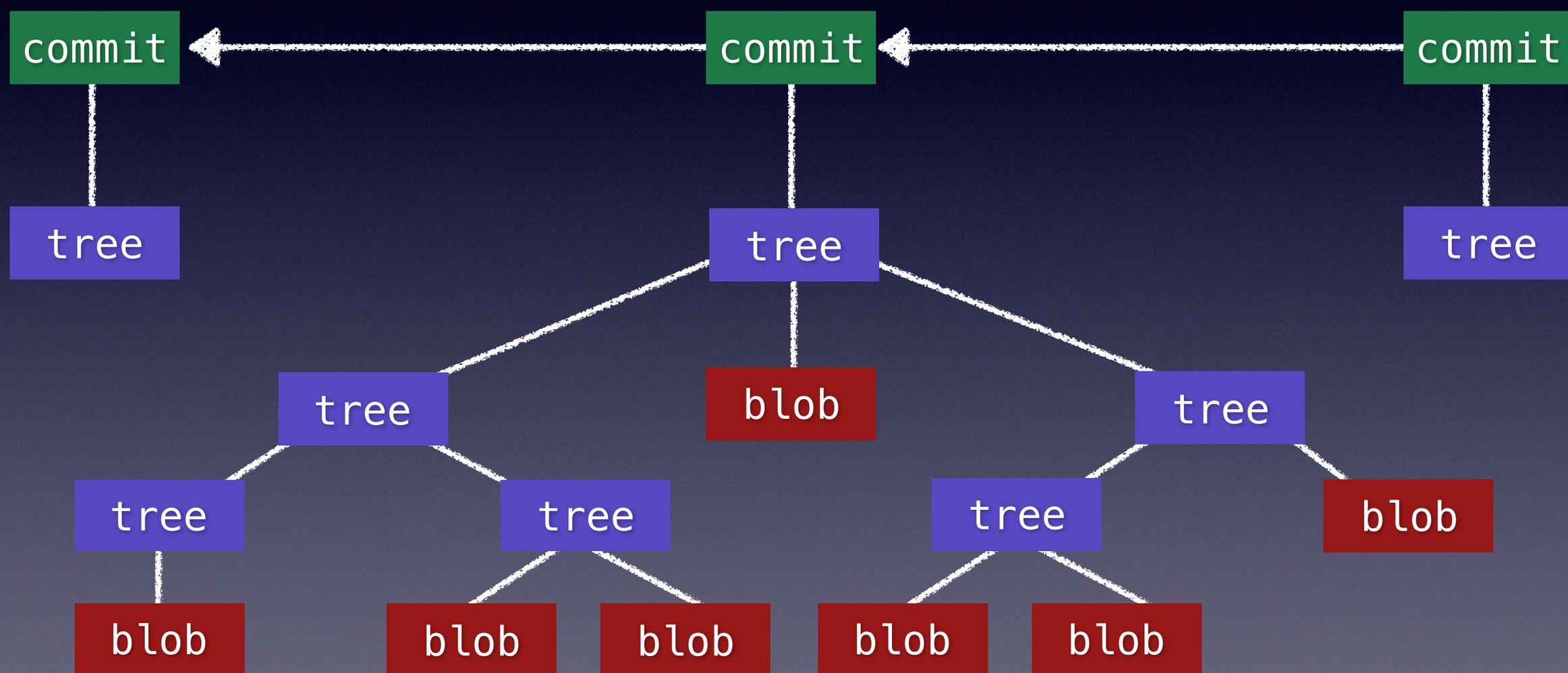
SIT objects

SIT objects

SIT objects



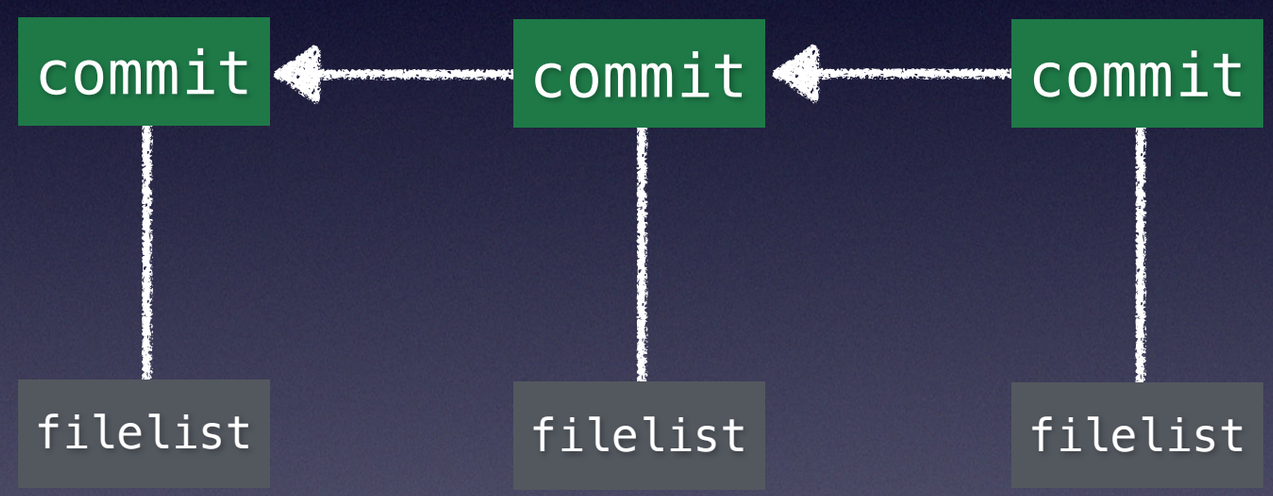
SIT objects


commit
commit
commit
tree
tree
tree
tree
blob
tree
tree
tree
blob
blob
blob
blob
blob
blob

SIT objects

- 为什么不直接用一个路径的列表?

filelist	size
5b1d3	ArrayList.h
cdc8b	ArrayList.cpp
1de87	test/test_list.cpp
3f724	test/test_map.cpp
d218c	test/test_set.cpp
c83d7	test/test.h
27c61	test/test.cpp
1de87	lib/jemalloc/ac.h
3f724	lib/jemalloc/ac.c



SIT objects

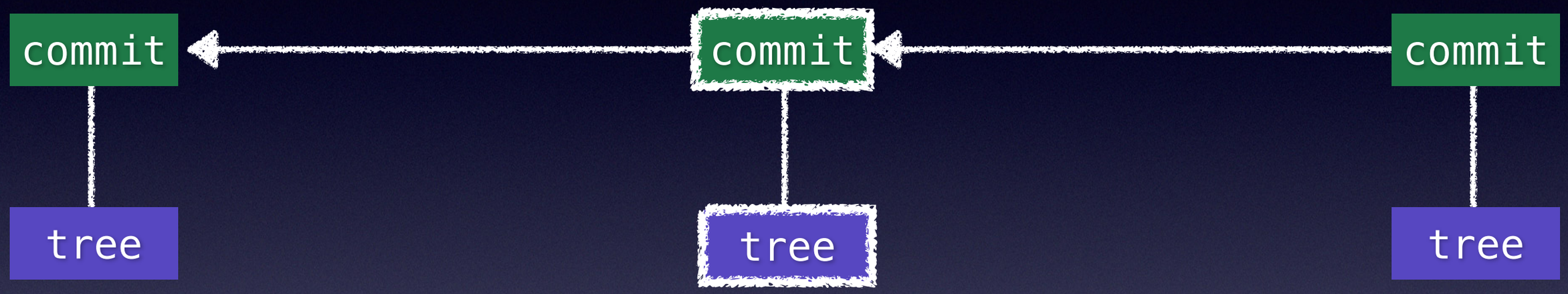
SHA1 (40bytes)	Path (22bytes)
43a3874f9aabf381df3b382abaab067a16182ded	ArrayList.cpp

- 假设 64字/行
- 假设有 65536 个文件
- 那么单次 commit 所需要空间为
- 64 * 65536 bytes = 4 MBytes
- Linux 有 500,000+ 的提交次数
- 光是保存 commit 记录就需要 2 GBytes

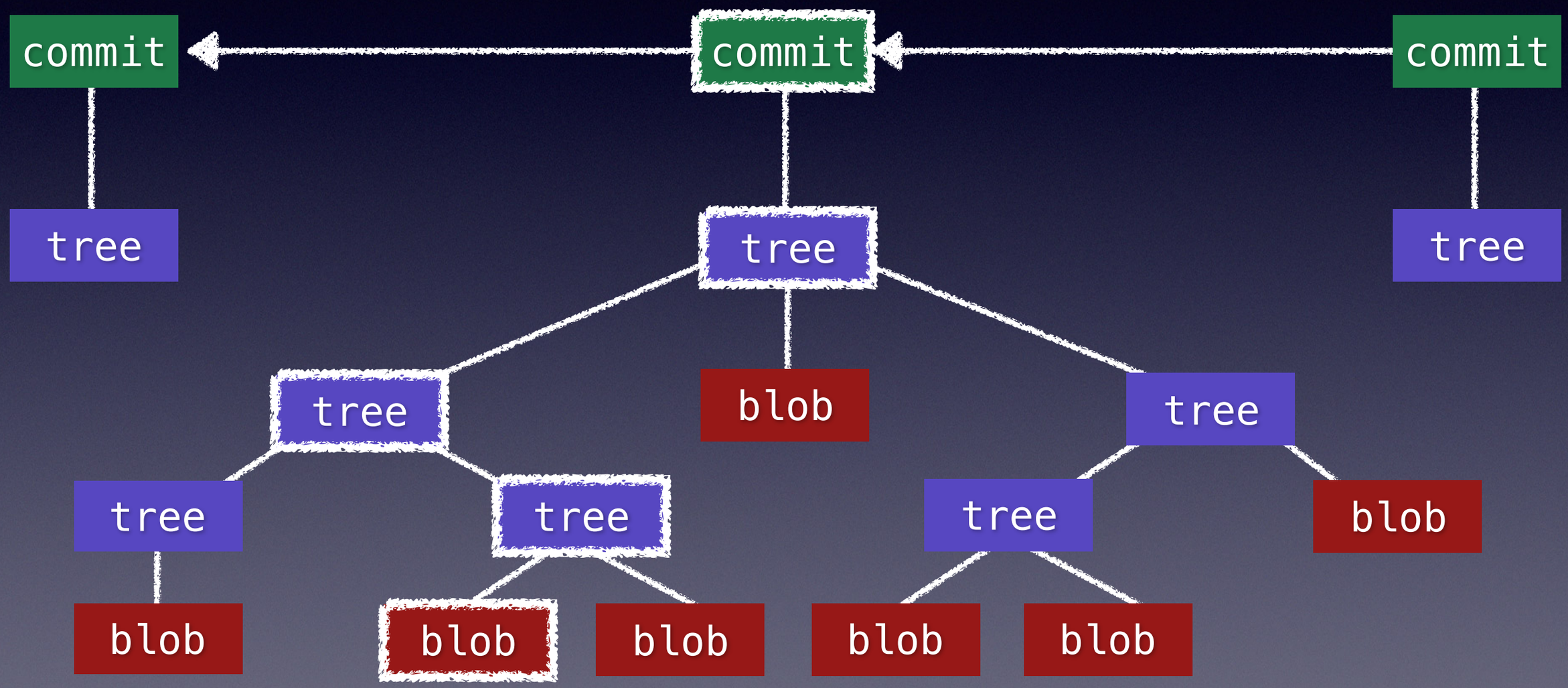
SIT objects

SIT objects

SIT objects



SIT objects


commit
commit
commit
tree
tree
tree
tree
blob
tree
tree
tree
tree
blob
blob
blob
blob
blob

diff

diff

```
this is the first line
this is the second line
hello world
hello world, again
```

```
this is the first line
why not insert a line?
this is the third line
hello world, again
```

```
 this is the first line
-this is the second line
+why not insert a line?
+this is the third line
-hello world
 hello world, again
```

diff

```
this is the first line   => 74
this is the second line  => 18
hello world              => 35
hello world, again       => 68
```

```
this is the first line   => 74
why not insert a line?   => 48
this is the third line   => 52
hello world, again       => 68
```

```
A: 74 18 35 68
B: 74 48 52 68
```

最长公共子序列
Longest Common Subsequence

diff

```
this is the first line    => 74
this is the second line   => 18
hello world               => 35
hello world, again        => 68
```

```
this is the first line    => 74
why not insert a line?    => 48
this is the third line    => 52
hello world, again        => 68
```

```
A: 74 18 35 68
B: 74 48 52 68
```

最长公共子序列
Longest Common Subsequence

diff

```
this is the first line    => 74
this is the second line   => 18
hello world               => 35
hello world, again        => 68
```

```
this is the first line    => 74
why not insert a line?    => 48
this is the third line    => 52
hello world, again        => 68
```

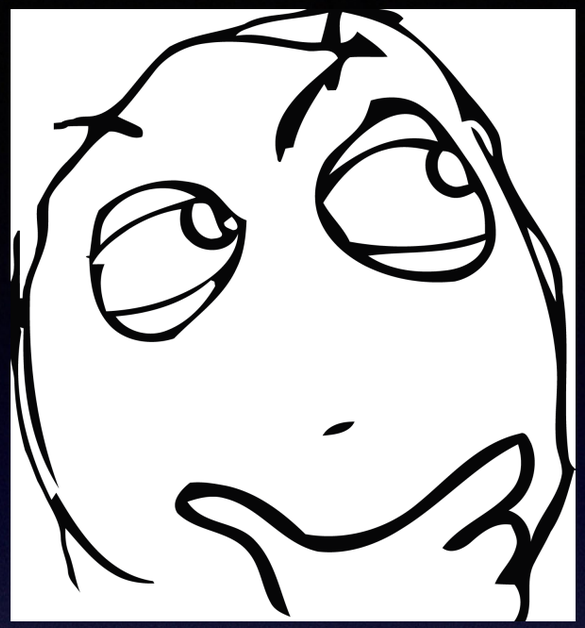
```
 this is the first line   => 74
-this is the second line  => 18
+why not insert a line?   => 48
+this is the third line   => 52
-hello world              => 35
 hello world, again       => 68
```

Hash

- `std::string line;`
- `size_t h = std::hash(line);`

- 32 bit: size_t == unsigned int
- 64 bit: size_t == unsigned long long

Hash



- `std::string line;`
- `size_t h = std::hash(line);`

- 32 bit: size_t == unsigned int

$$100\% - \prod_{k=1}^{50000} \left(1 - \frac{k-1}{2^{32}}\right) = 25.3\%$$

Hash

- **std::hash** => MurmurHash2
  - **std::string** => 32bit / 64bit

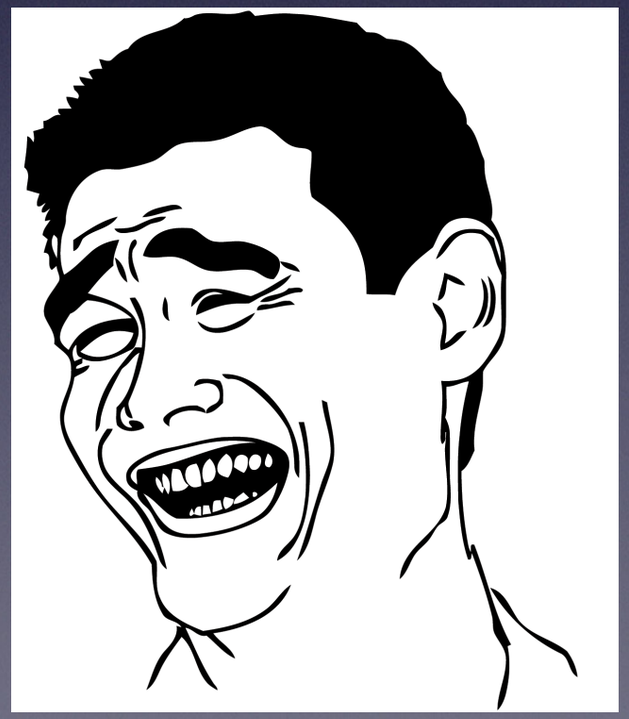
- **sit::hash** => MurmurHash3
  - **std::string** => 128bit
  - faster than **std::hash**

LCS

- A[N], B[M]
- Longest Common Subsequence?
- Dynamic Programming
  - Time: O(N*M)
  - Space: O(N*M)

LCS

- A[N], B[M]
- Longest Common Subsequence?
- ~~Dynamic Programming~~
  - ~~Time: O(N*M)~~
  - ~~Space: O(N*M)~~

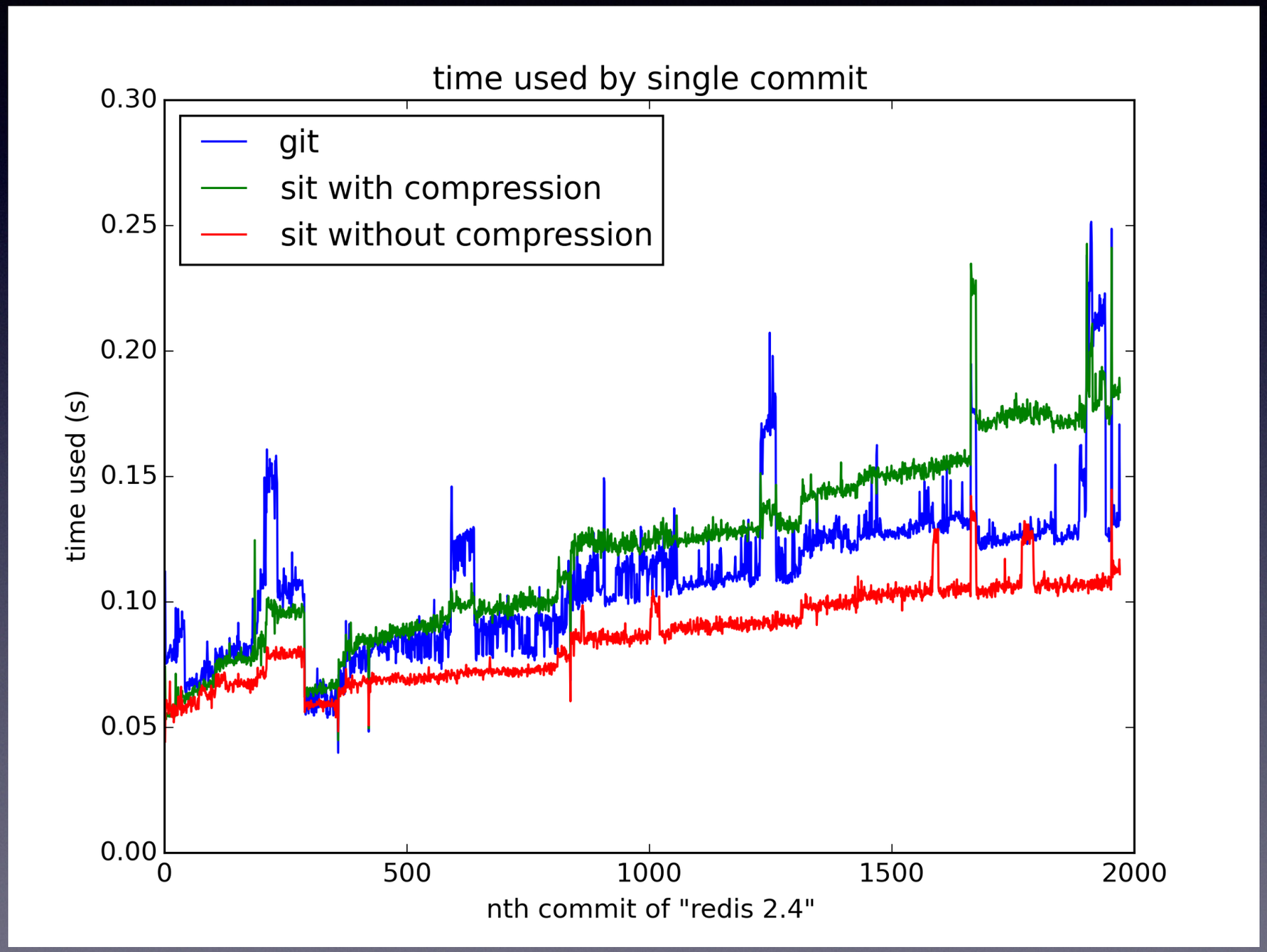


LCS

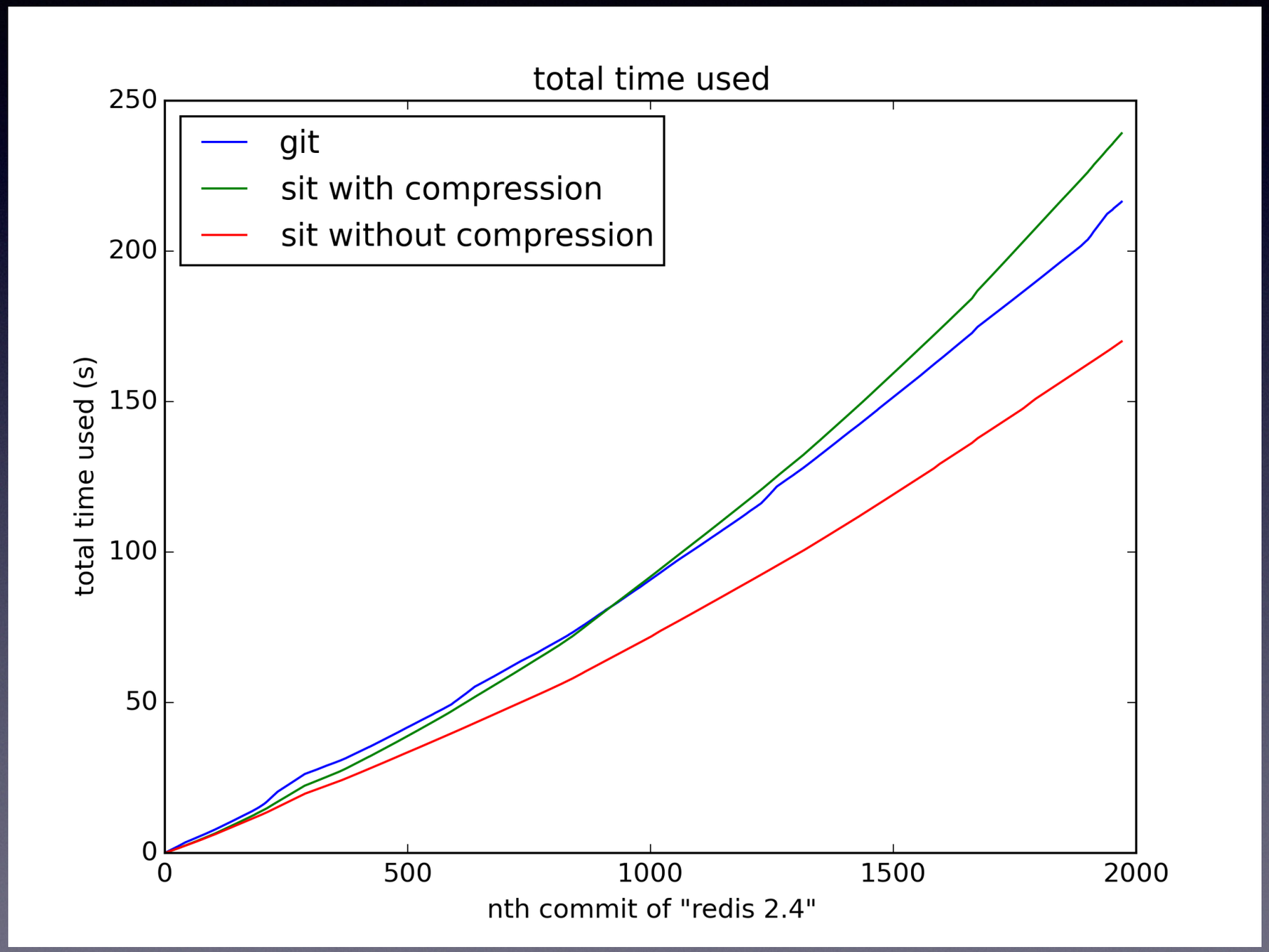
- *An O(ND) Difference Algorithm and its Variations* by Eugene Myers
  - Time:
    - Expected: O(N+M+D^2)
    - Worst: O((N+M) * D)
    - Optimized: O((N+M)log(N+M) + D^2)
  - Space: O(N+M)

SIT的效率?

SIT vs GIT


time used by single commit
git
sit with compression
sit without compression
time used (s)
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0
500
1000
1500
2000
nth commit of "redis 2.4"

SIT vs GIT


total time used
250
200
150
100
50
0
0
500
1000
1500
2000
total time used (s)
nth commit of "redis 2.4"
git
sit with compression
sit without compression

SIT vs GIT

Space Usage

Benchmark Repo: Redis 2.4		
.GIT	94,324 kb	100%
.SIT (with compression)	82,180 kb	87.1%
.SIT (without compression)	264,004 kb	279.9%

SIT的…?

SIT的代码量

Developer — abcdabcd987@clq-rmbp: ~/Developer — ~/Developer — zsh...

```
→  Developer  cloc sit/
      63 text files.
      61 unique files.
      38 files ignored.

http://cloc.sourceforge.net v 1.62  T=0.25 s (105.5 files/s, 13957.7 lines/s)
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
C++                             12            297             62           2164
C/C++ Header                    11            132            290            298
Python                           1             21             12            117
make                             2             10              0             36
-------------------------------------------------------------------------------
SUM:                            26            460            364           2615
-------------------------------------------------------------------------------
→  Developer
```

SIT的用到的技术

- C++ 11
- Boost
- MurmurHash3
- zlib
- python 3 (for stress testing)

SIT

- Author
  - 陈乐群
  - 柯嵩宇
- Source Code (Open Sourced)
  - https://github.com/abcdabcd987/sit

Thank You!