

散列

散列函数：随机数

又由谁去体现这世间的幸福、骄傲和快乐？只好听凭偶然，是没有道理好讲的

不好，不好！我这里有一方手帕，你顶在头上，遮了脸，撞个天婚，
教我女儿从你跟前走过，你伸开手扯倒那个就把那个配了你罢

天下就没有偶然，那不过是化了妆的、戴了面具的必然

邓俊辉

deng@tsinghua.edu.cn

(伪) 随机数法

❖ 循环: $\text{rand}(x + 1) = [a \times \text{rand}(x)] \% M$ //M素数, $a \% M \neq 0$

$$a = 7^5 = 16,807 = \boxed{100000110100111}_b$$

$$M = 2^{31} - 1 = 2,147,483,647 = 01111111 \boxed{11111111} 11111111 \boxed{11111111}_b$$

❖ 径取: $\text{hash}(\text{key}) = \text{rand}(\text{key}) = [\text{rand}(0) \times a^{\text{key}}] \% M$

种子: $\text{rand}(0) = ?$

❖ 把难题推给伪随机数发生器, 但是...

❖ (伪) 随机数发生器的实现, 因具体平台、不同历史版本而异

创建的散列表可移植性差——故需慎用此法!

(伪) 随机数法: The C Programming Language (2nd edn), p46

❖ `unsigned long int next = 1; //sizeof(long int) = 8`

`void srand(unsigned int seed) { next = seed; } //sizeof(int) = 4 or 8`

`int rand(void) { //1103515245 = $3^5 * 5 * 7 * 129749$`

`next = next * 1103515245 + 12345; //M + A + ...`

`return (unsigned int) (next/65536) % 32768; //...D`

}

next



rand



❖ `int rand() { int uninitialized; return uninitialized; }`

`char* rand(t_size n) { return (char*) malloc(n); }`

就地随机置乱：任给一个数组A[0, n), 理想地将其中元素的次序随机打乱

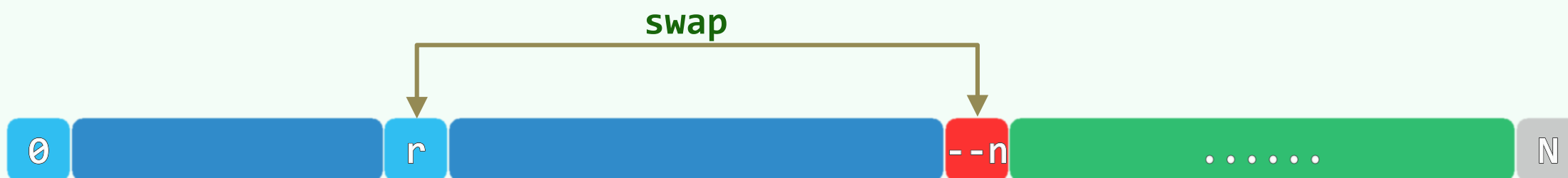
//[R. Fisher & F. Yates, 1938], [R. Durstenfeld, 1964], [D. E. Knuth, 1969]

```
void shuffle( int A[], int n ) {
```

```
    for ( ; 1 < n; --n ) //自后向前, 依次将各元素
```

```
        swap( A[ rand() % n ], A[ n - 1 ] ); //与随机选取的某一前驱 (含自身) 交换
```

```
    } //20! < 264 < 21!
```



❖ 的确可以等概率地生成所有 $n!$ 种排列? $20! < 2^{64} < 21!$