串

Karp-Rabin算法：字宽

13-F3

God kisses the finite in his love and man the infinite.

当一个人反复思考的时候，就必定会出现悖论，然而不管你们会怎么说，
我宁愿做一个持有悖论的人，也不愿做心存偏见的人

邓俊辉
deng@tsinghua.edu.cn

$power_a(n) = a^n$

- **平凡实现**:

```
pow = 1; //O(1)
while ( 0 < n ) //O(n)
  { pow *= a; n--; } //O(1+1)
```

- $T(n) = 1 + 2n = \mathcal{O}(n)$

线性? 伪线性!

- **所谓输入规模，准确地应定义为**

  用以描述输入所需的空间规模

- **对于此类数值计算**

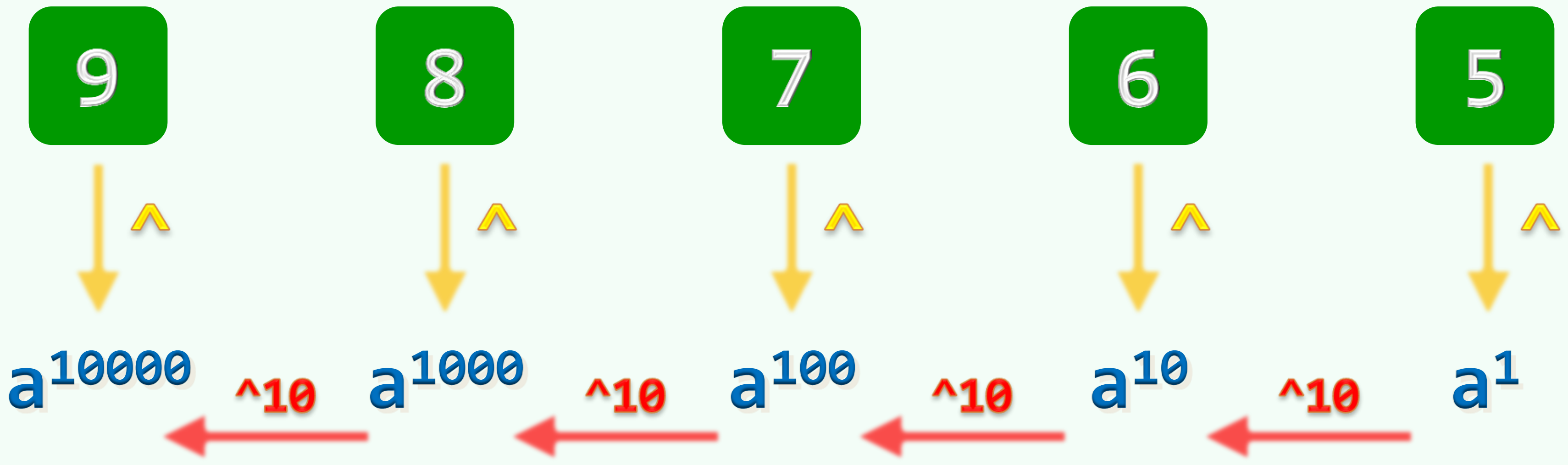
  即是n的二进制位数，亦即n的打印宽度

  $r = \lceil \log_2(n+1) \rceil = \mathcal{O}(\log n)$

  $T(r) = \mathcal{O}(2^r)$ //指数复杂度!

a^{98765}

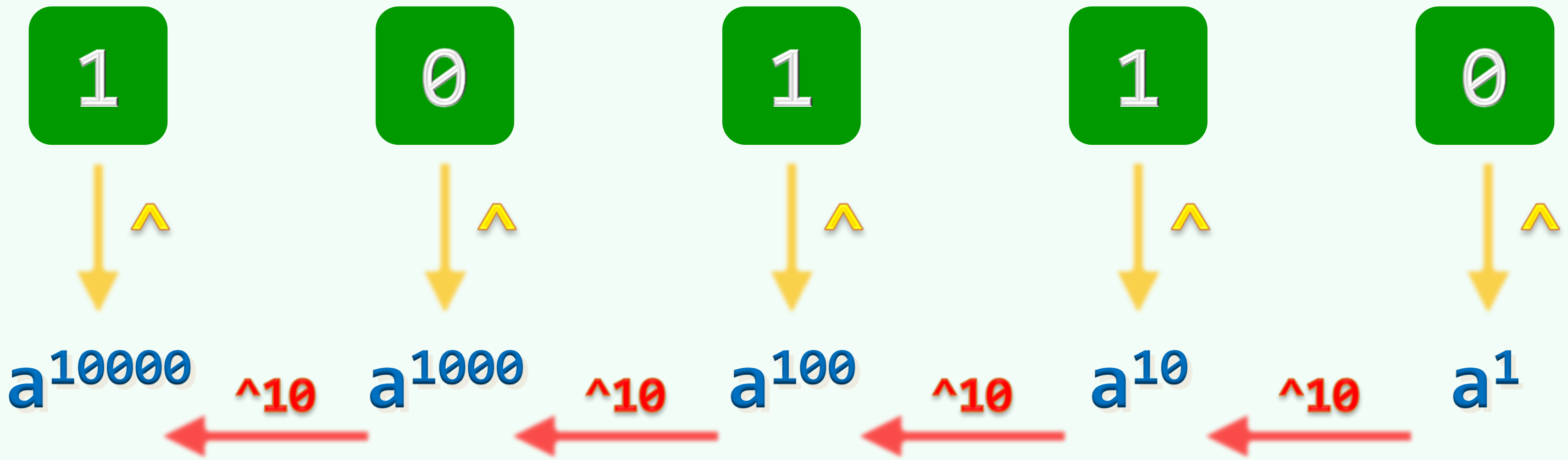
$$a^{9\cdot 10^4 \;+\; 8\cdot 10^3 \;+\; 7\cdot 10^2 \;+\; 6\cdot 10^1 \;+\; 5\cdot 10^0}$$



$$\left(a^{10^4}\right)^9 \cdot \left(a^{10^3}\right)^8 \cdot \left(a^{10^2}\right)^7 \cdot \left(a^{10^1}\right)^6 \cdot \left(a^{10^0}\right)^5$$

a^{10110b}

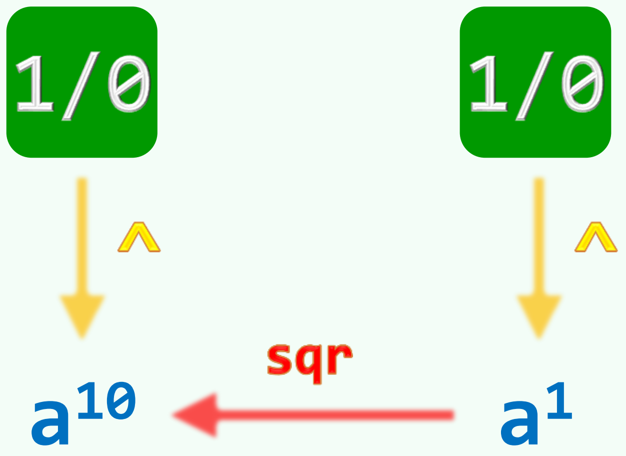
$$a^{1\cdot 2^4 \;+\; 0\cdot 2^3 \;+\; 1\cdot 2^2 \;+\; 1\cdot 2^1 \;+\; 0\cdot 2^0}$$



$$\left(a^{2^4}\right)^1 \cdot \left(a^{2^3}\right)^0 \cdot \left(a^{2^2}\right)^1 \cdot \left(a^{2^1}\right)^1 \cdot \left(a^{2^0}\right)^0$$

从 $\mathcal{O}$(n) 到 $\mathcal{O}$(r = logn)

```
int power( int a, int n ) {
   int pow = 1, p = a; //O(1 + 1)
   while (0 < n) { //O(logn)
      if (n & 1) //O(1)
         pow *= p; //O(1)
      n >>= 1; //O(1)
      p *= p; //O(1)
   }
   return pow; //O(1)
}
```



- 输入规模 $= r = \lceil \log_2 (n+1) \rceil$
- 运行时间 $= T(r) = 1 + 1 + 4r + 1 = \mathcal{O}(r)$
- 如此，“实现”了从指数到线性的改进

悖论?

❖ 根据以上算法，“可以”在 $\mathcal{O}(\log n)$ 时间内计算出 $power(n) = a^n$

❖ 然而，a^n 的二进制展开宽度为 $\Theta(n)$

这意味着，即便是直接打印 a^n，也至少需要 $\Omega(n)$ 时间......哪里错了?

❖ 类似的悖论对 $fib(n)$ 也存在...

❖ 令：$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} fib(0) & fib(1) \\ fib(1) & fib(2) \end{bmatrix}$，则：$A^n = \begin{bmatrix} fib(n-1) & fib(n) \\ fib(n) & fib(n+1) \end{bmatrix}$

❖ 因此参照上述power()算法，似乎也“可以”在 $\mathcal{O}(\log n)$ 时间内计算出 $fib(n)$

❖ RAM模型：常数代价准则 (uniform cost criterion)

对数代价准则 (logarithmic cost criterion)