



1. 强制进退



同归测



可以的

(a)、(b) 对比：(b) 中敌人移动：领域移动
目标：找到敌人

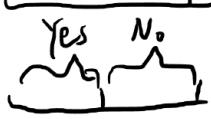
解：1. 设计算法

2. 设计耗时最短

② Day 1:



Day 2:



Day x:



End

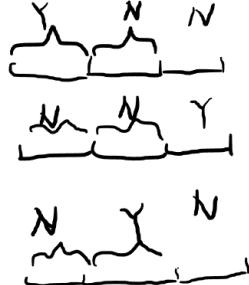
$\log_2 n$ 天

有没有改进？如果每天测 1 次（不是 2 次）

我仍旧可以实现 ②

为什么题目给 2 次？

③

3种情况 区间变为 $\frac{1}{3}$

$$\log_3 n$$

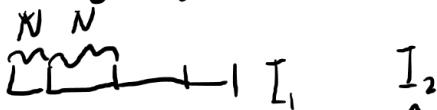
有没有改进？

3.1



$$\frac{3}{8} \quad \frac{3}{8} \quad \frac{2}{8}$$

3.2.

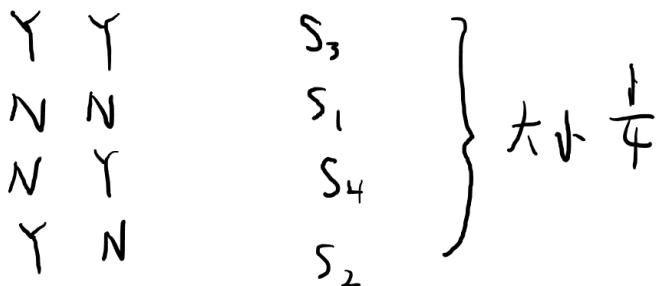


④



$$S = [1:n]$$

$$I_1, VI_2 \text{ takes } \frac{3}{4} \quad S \setminus (I_1, VI_2) \text{ takes } \frac{1}{4}$$

 I_1, I_2 T-步范围

$$I_1 \cap I_2 \neq \emptyset, (Y, Y) \text{ 成立}$$

Algo.4 takes $\lceil \log_4 n \rceil$

有没有改进? I_1, I_2 2个 boolean variable
 $(I_1, I_2) \rightarrow$ T-步范围 最多有 4 个情况 (xx)
不能再改进.

(b) S_d : 范围在 d-th 使用 Algo.4.

$$S_1 = [1:n] \quad x$$

$$S_2 = [l_2, r_2]$$

$$S_d = [l_d, r_d] : l_d = r_d \quad \text{1-st Day}$$



2-nd Day

$$\text{Expand}(S_2) = \{x \in [1, n] \mid \exists y \in S_2, |x - y| \leq 1\}$$

$$\text{记 } S_2 = [l_2, r_2] = [l_2 - 1, r_2 + 1] \quad \text{更严谨}$$

What if $l_2 = 0$? $r_2 = n$?

在 Algo.4 基础上, 在迭代中尾作一次扩展.

得到了(b) 的方法，记为 Algo.5.

Algo.5 时间? $\lceil \log_4(n+2\log_4 n) \rceil$



assume $m \leq n$



目标: A+B sorted



① 简单方法: $O(m+n)$ combine and return k-th.

not $O(\log k)$



$k \leq m$?



$k \leq n$? $B[1, \dots, n]$

Recursive Algorithm

KTH(A[1..m], k)

base case: ① $m=0$

return $B[k]$

② $n=0$

return $A[k]$

③ $k=1$ return $\min(A_1, B_1)$

④ $k=2$ return $\max(A_1, B_1)$ X

i 當 $A[i-i] \leftarrow \min\{m, \frac{k}{2}\}$ 之後 i, j , 及 $\frac{k}{2}$ 元素
j 當 $B[1-j] \leftarrow \min\{n, \frac{k}{2}\}$ start from $k = i+j$

split index

if $A[i] < B[j]$:

Go to KTH ($A[i+1, \dots m]$, $B[1 \dots n]$, $k-i$)

else: ($A[i] \geq B[j]$)

Go to KTH ($A[1 \dots m]$, $B[j+1, \dots n]$, $k-i$)

$\log k$ steps to trivial case.

if $A[i] < B[j]$, $A \cup B$ sorted

\exists at least i elements in $A \cup B$ s.t. $\leq A[i]$

要找 k -th element not in $A[i]$

if k -th element in $A[i]$ $|A[i]| = \frac{k}{2}$

$$|B[1-j]| = \frac{k}{2}$$



$$|A[i]| = |B[1 \dots j]| = \frac{k}{2} \quad \max(A[i], B[j])$$

?

3. bubble sort suffix in increasing order

a) $A[n-i+1] \dots A[n-1]$ largest elements

b) $A[j] = \max\{A[1], A[2], \dots, A[j]\}$

