Minerals 解説

tozangezan

問題概要

- N 種類の鉱物がそれぞれ2個ずつ、合計2N 個ある
- 以下の操作を1000000回以内行ってどれとどれが同じ鉱物かを特定せよ

- 操作: 装置に1個入れる、もしくは1個出す。その後、装置に入っている鉱物の種類数がわかる

制約

N ≤ 100

N ≤ 15000, 切片1～N は全部別の種類

N ≤ 15000

N ≤ 38000

N ≤ 39000

N ≤ 40000

N ≤ 41000

N ≤ 42000

N ≤ 43000

小課題 1

- 何回かけてもいいのでペアを見つけられますか？
- ${}_{2N}C_2$ 個のペアについて以下を行う
- 1. あるペアを装置に入れる
- 2. 装置が 1 を返したら同じ種類
- 3. ペアを装置から出す

- $O(N^2)$ 回装置を使う必要がある
- 5点

制約

N ≤ 100

N ≤ 15000, 切片1～N は全部別の種類

N ≤ 15000

N ≤ 38000

N ≤ 39000

N ≤ 40000

N ≤ 41000

N ≤ 42000

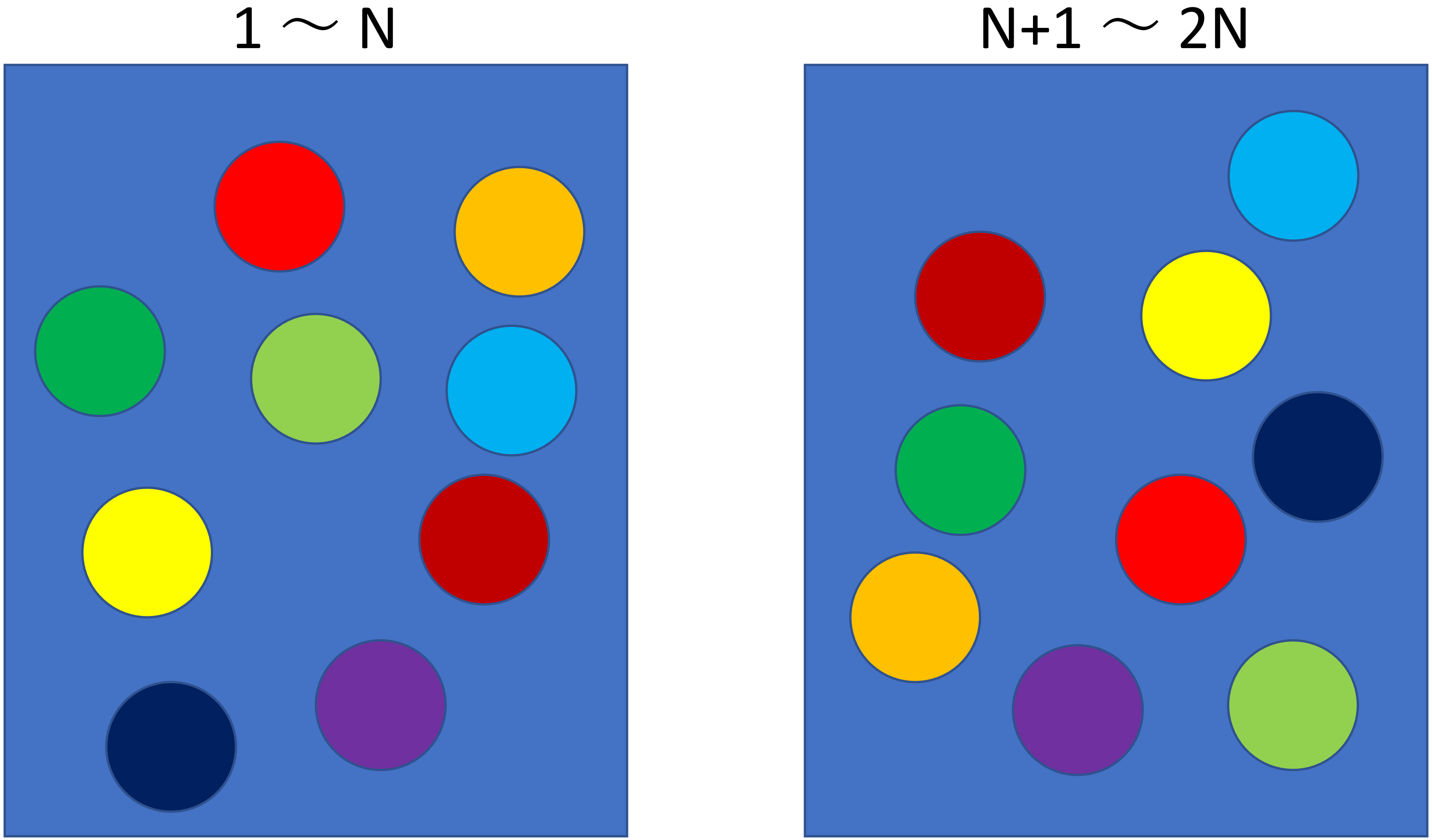
N ≤ 43000

小課題 2

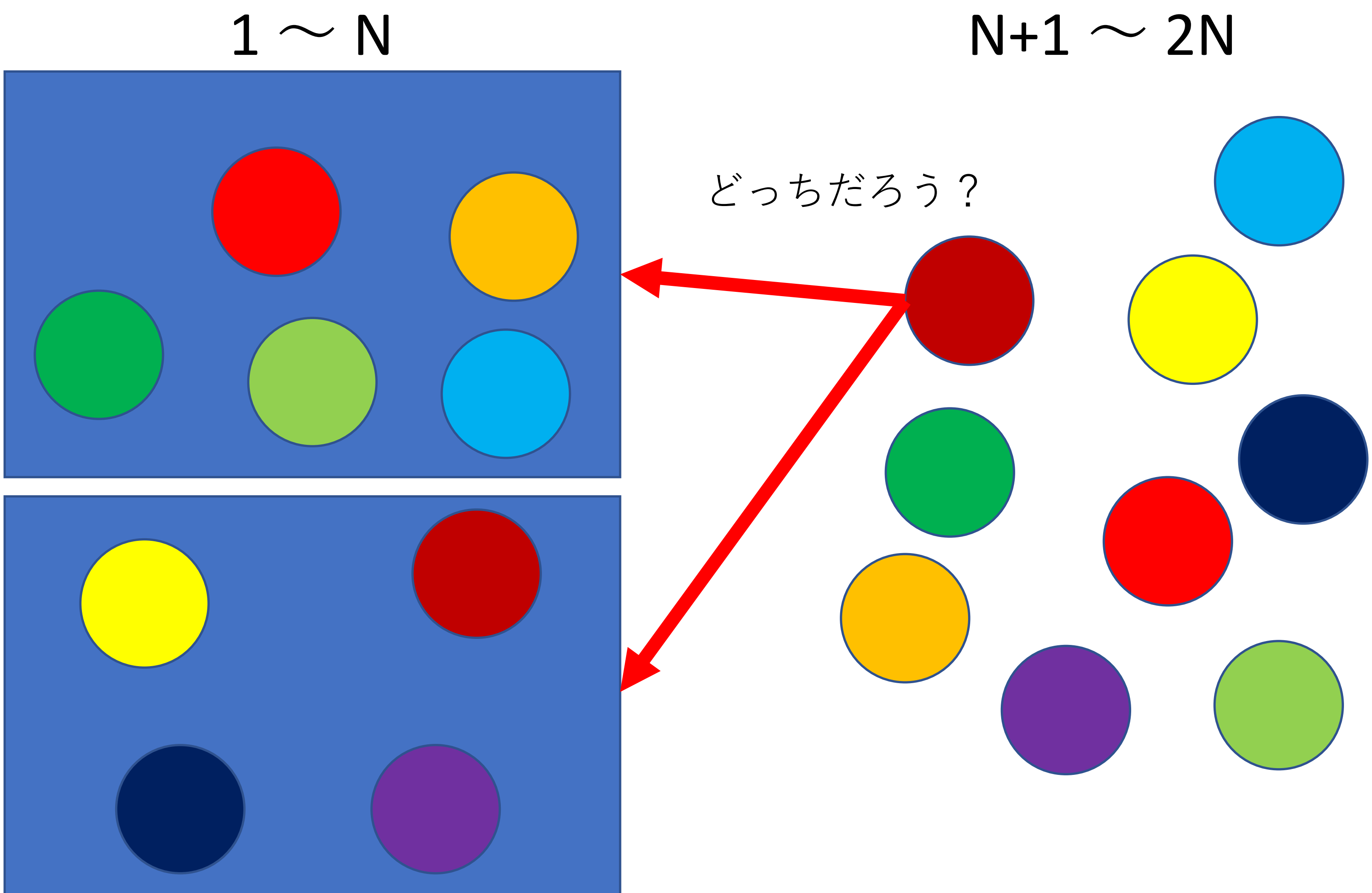
- N ≤ 15000
- O(N log N) なら何でも通るかな…？

- 切片1～N は全部別の種類
- どういうことだろう？

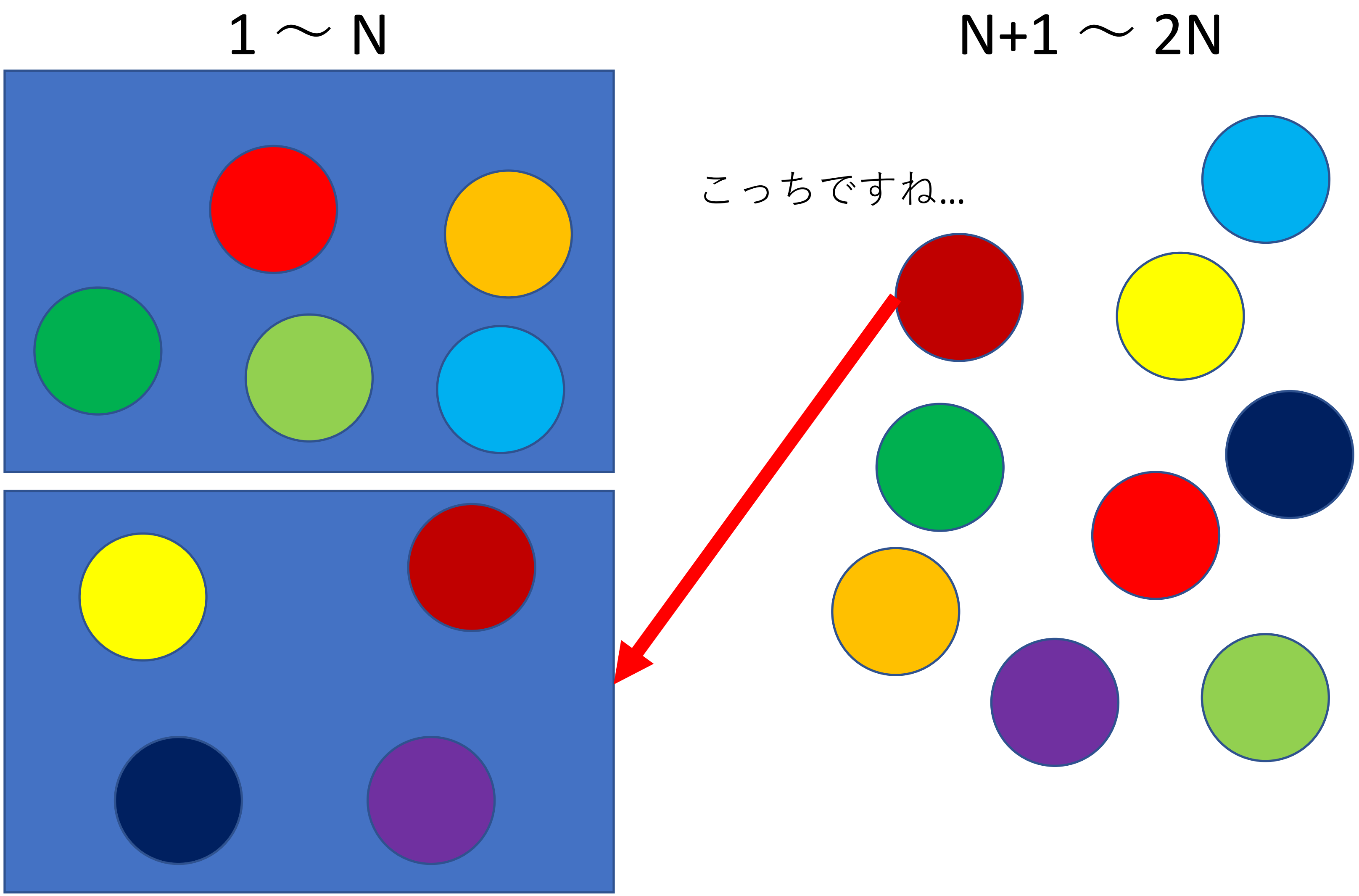
こういうことですね



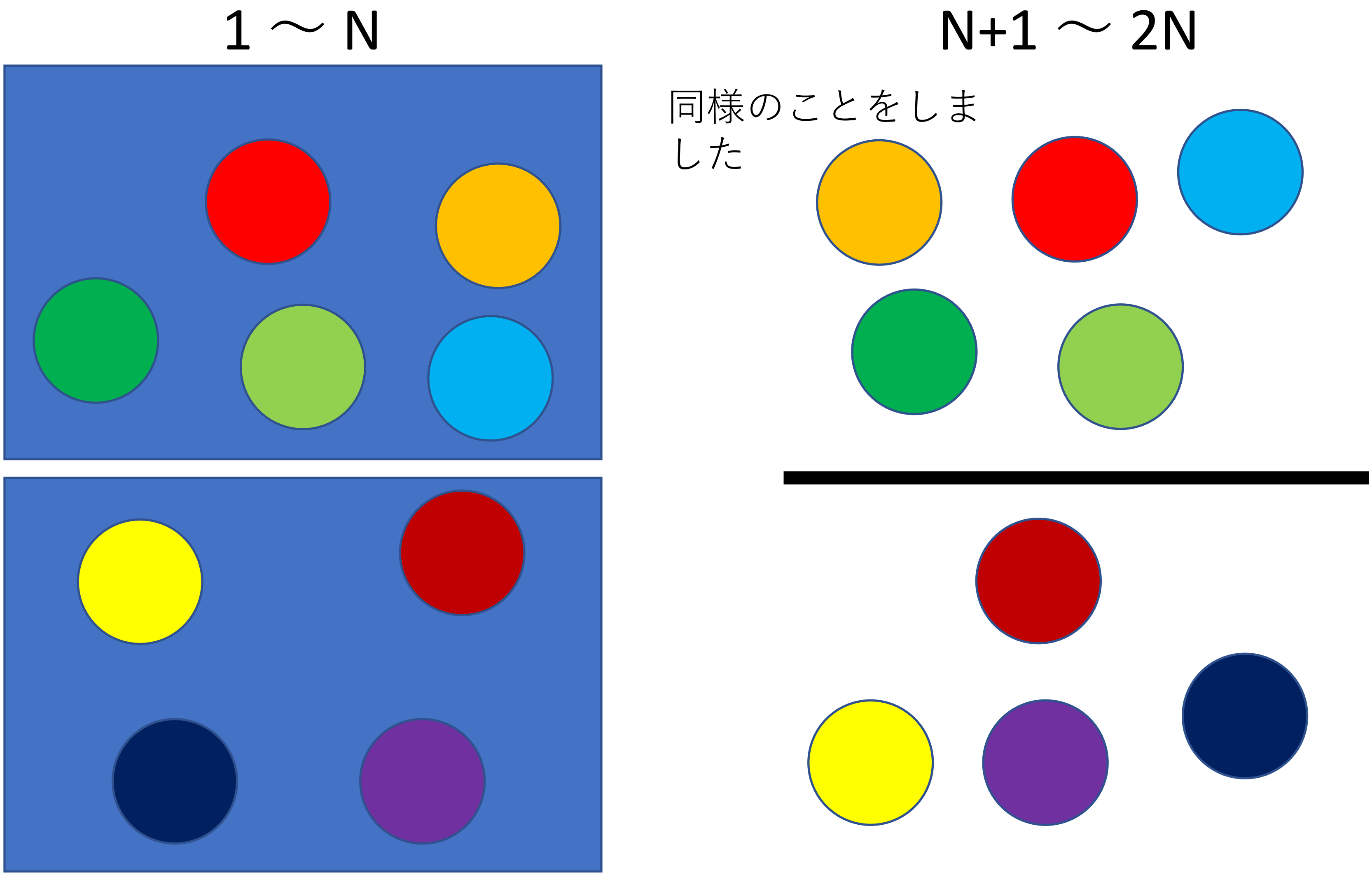
こういうことですね



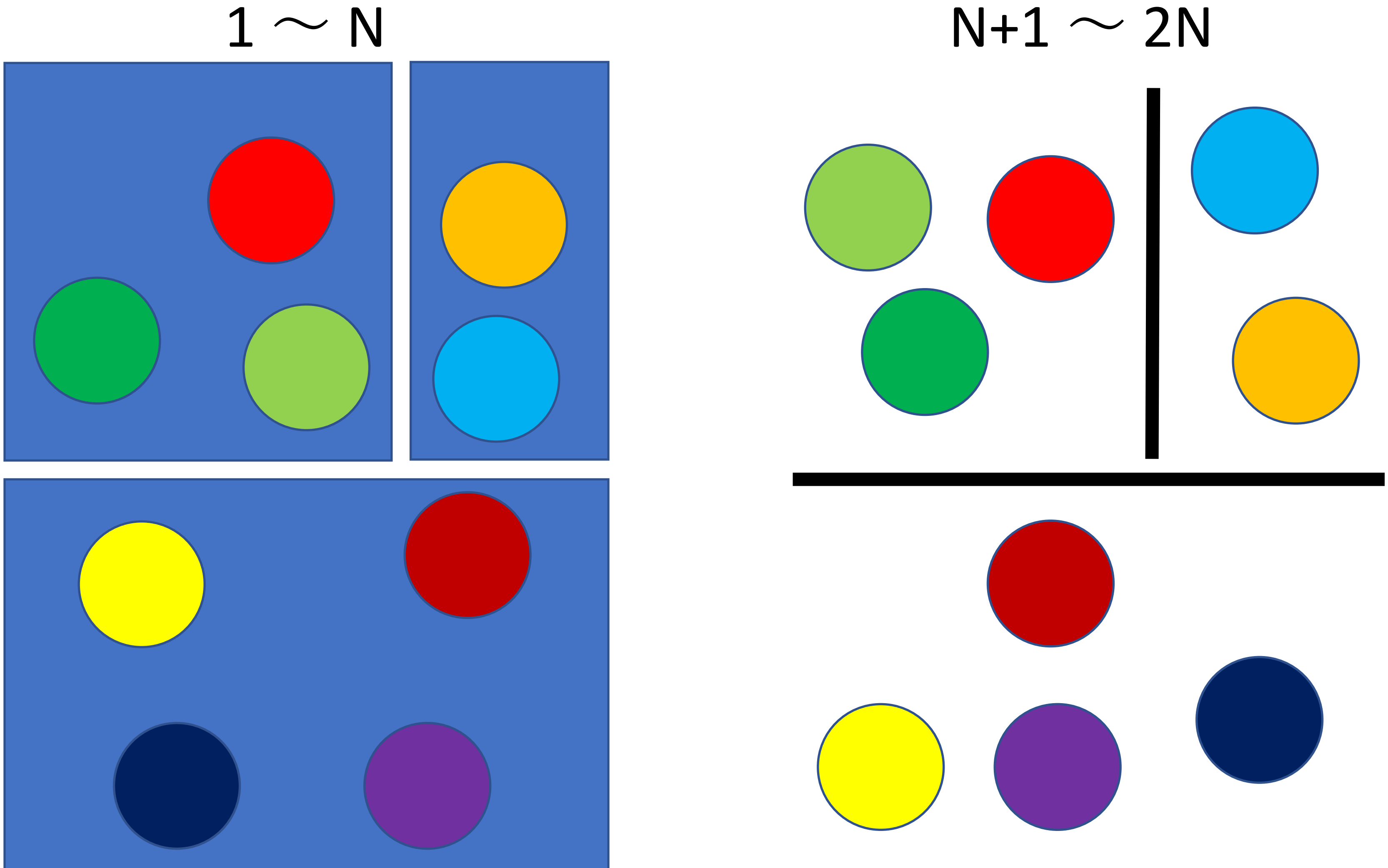
こういうことですね



こういうことですね

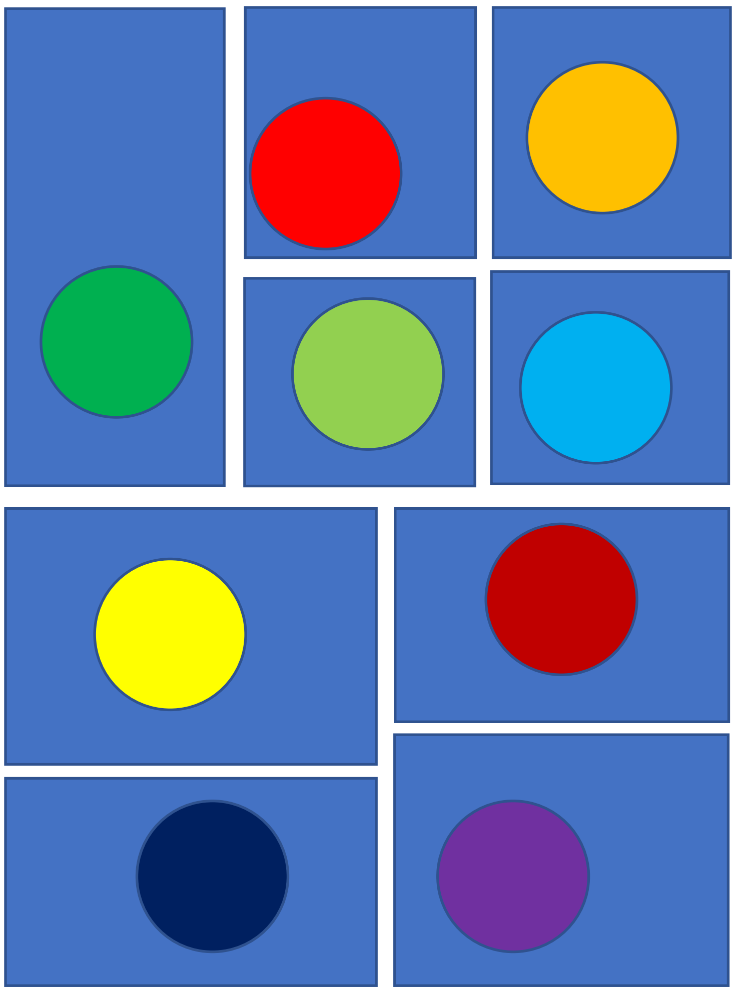


これは… 分割統治

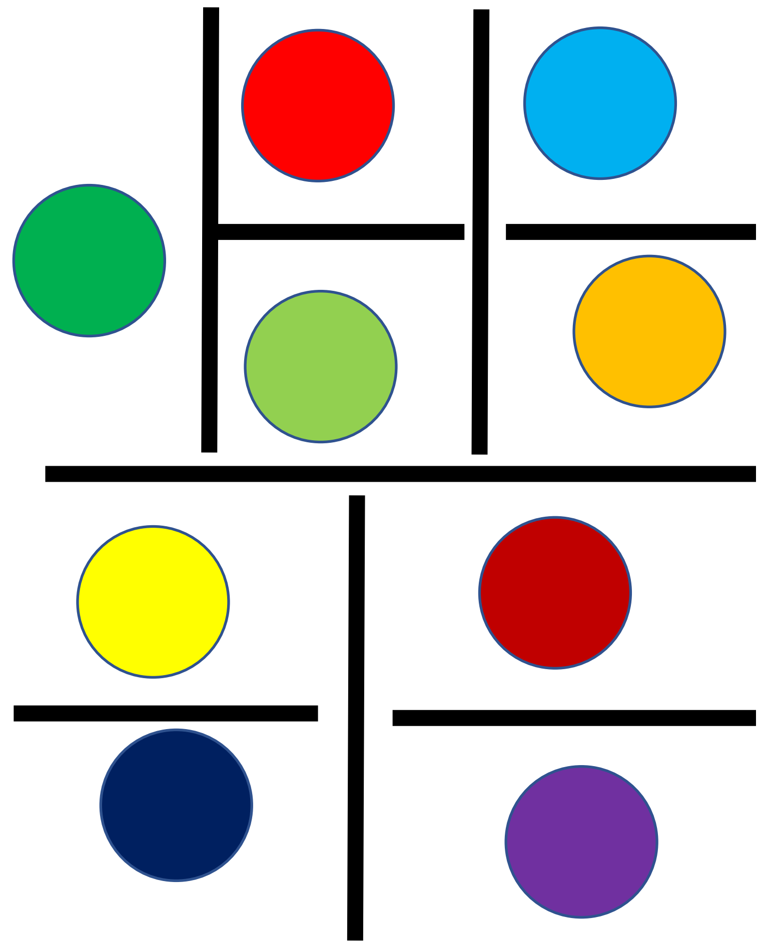


これは… 分割統治

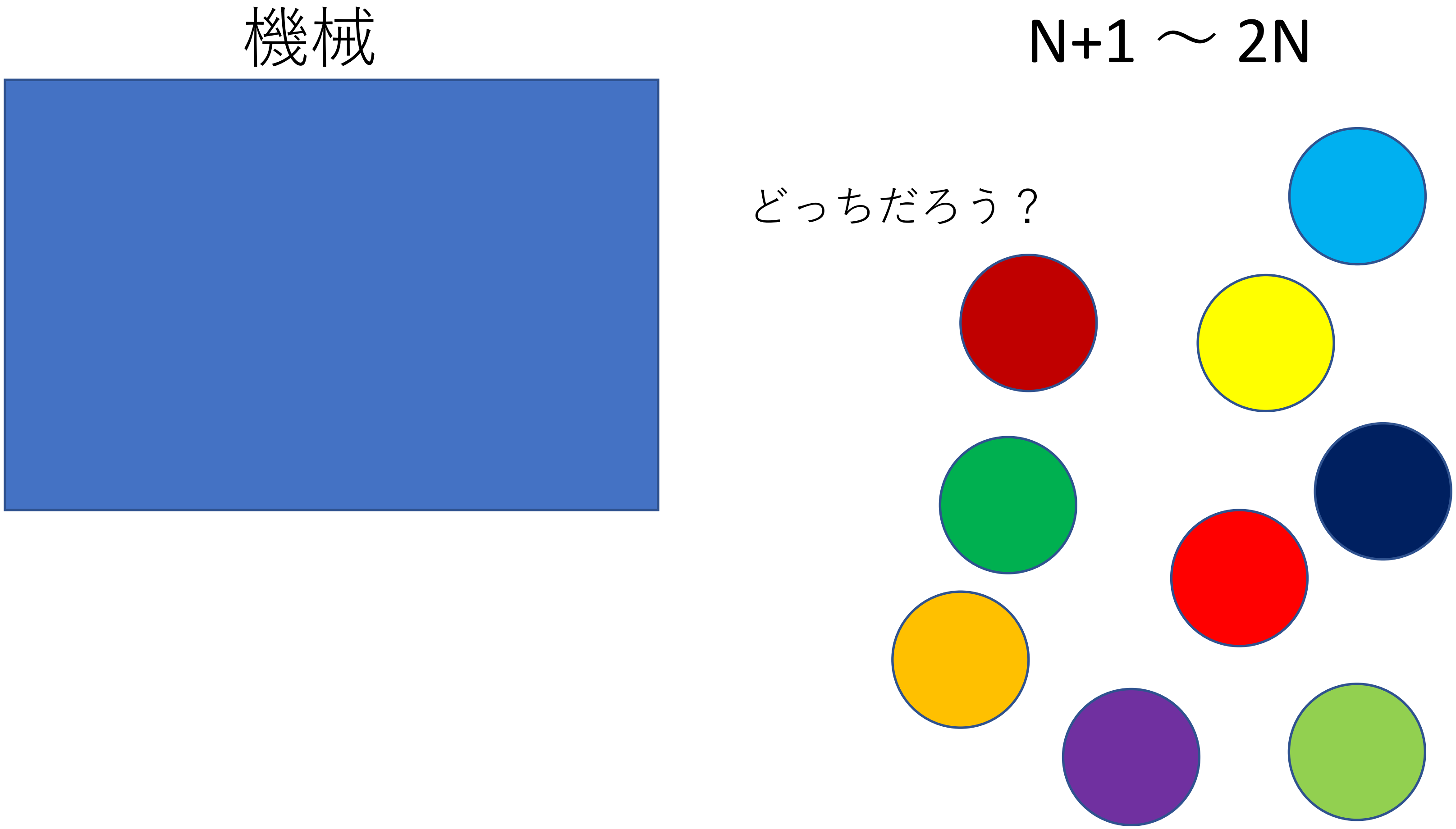
1 〜 N



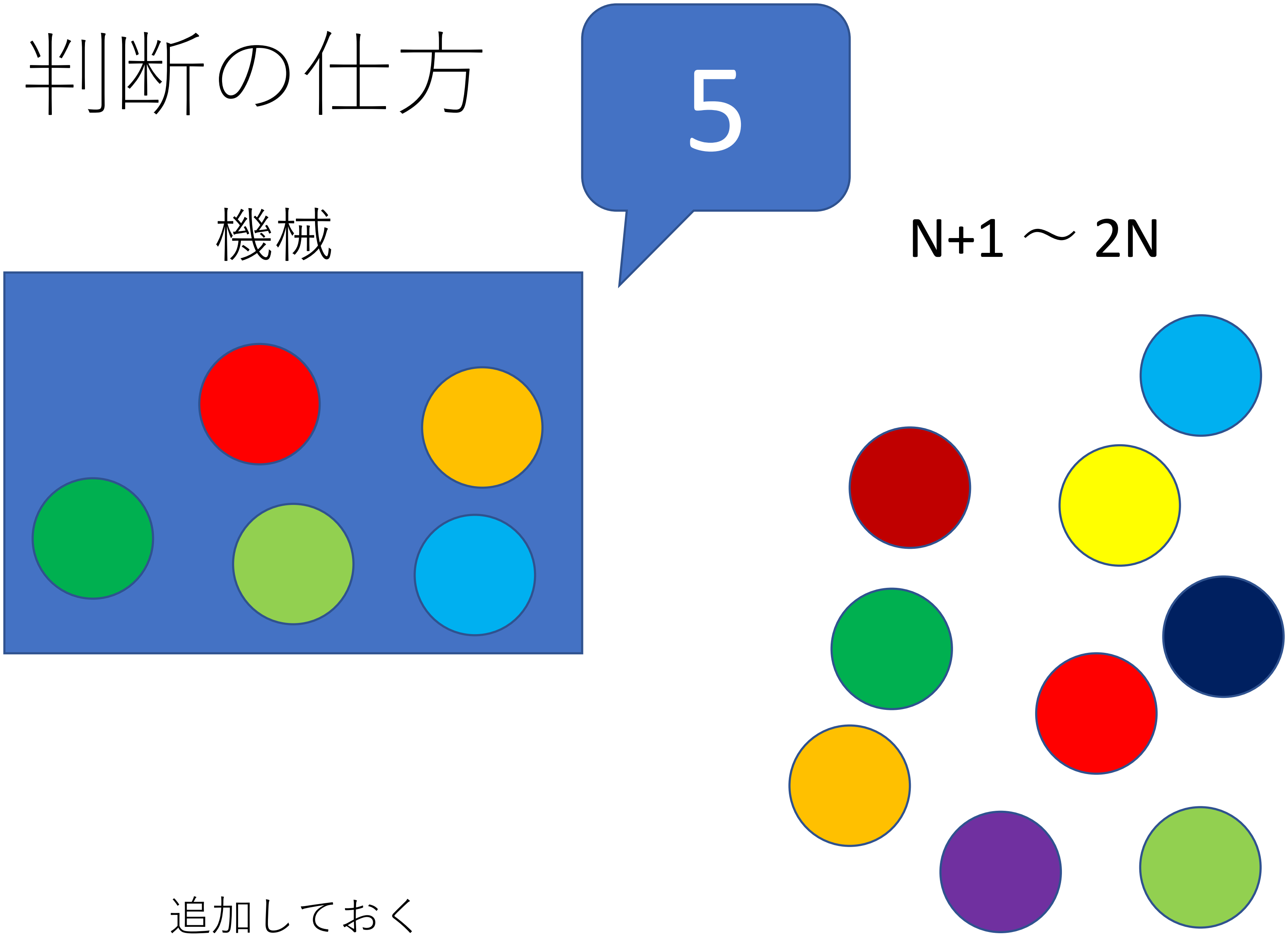
N+1 〜 2N



判断の仕方

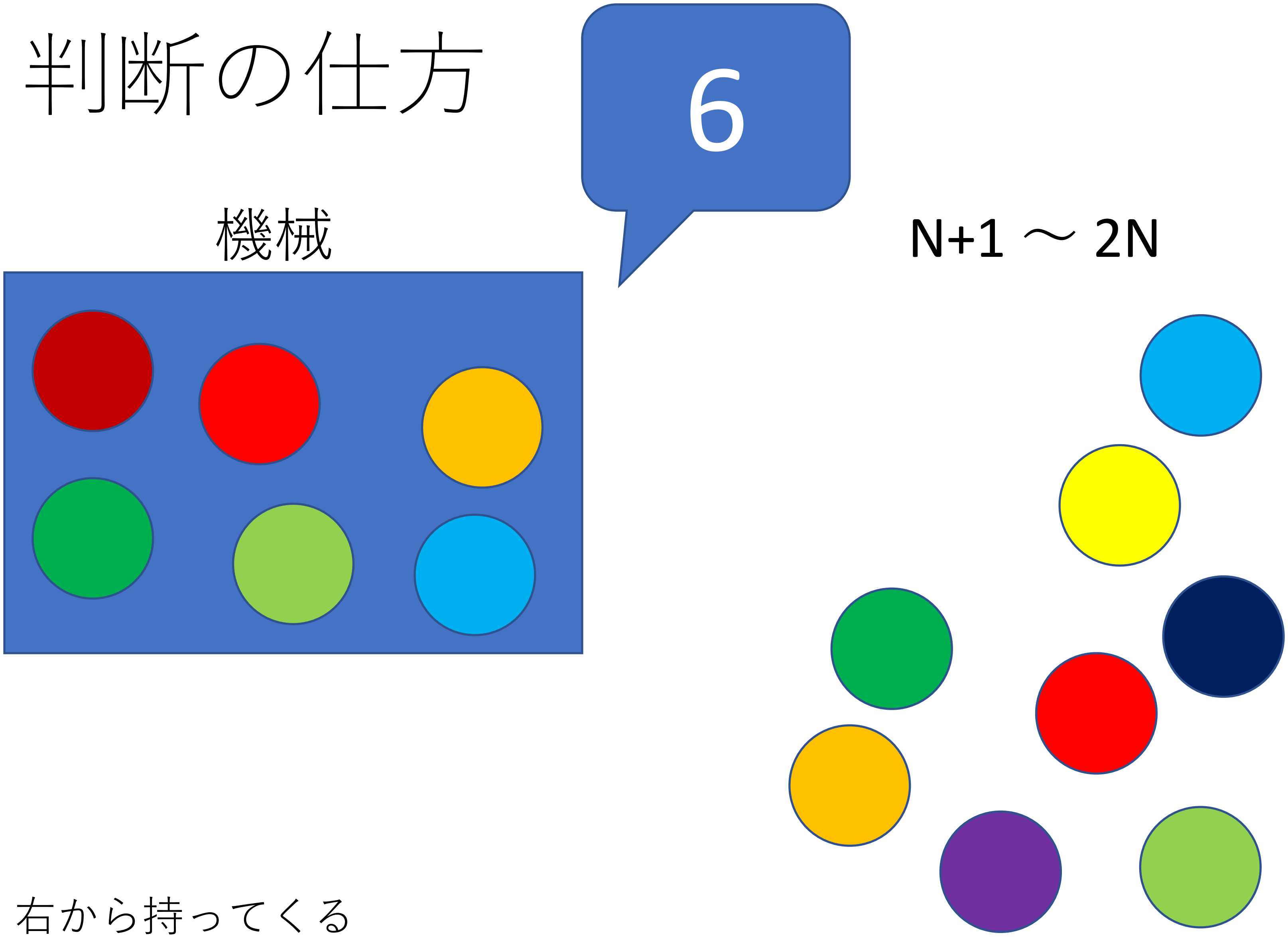


判断の仕方



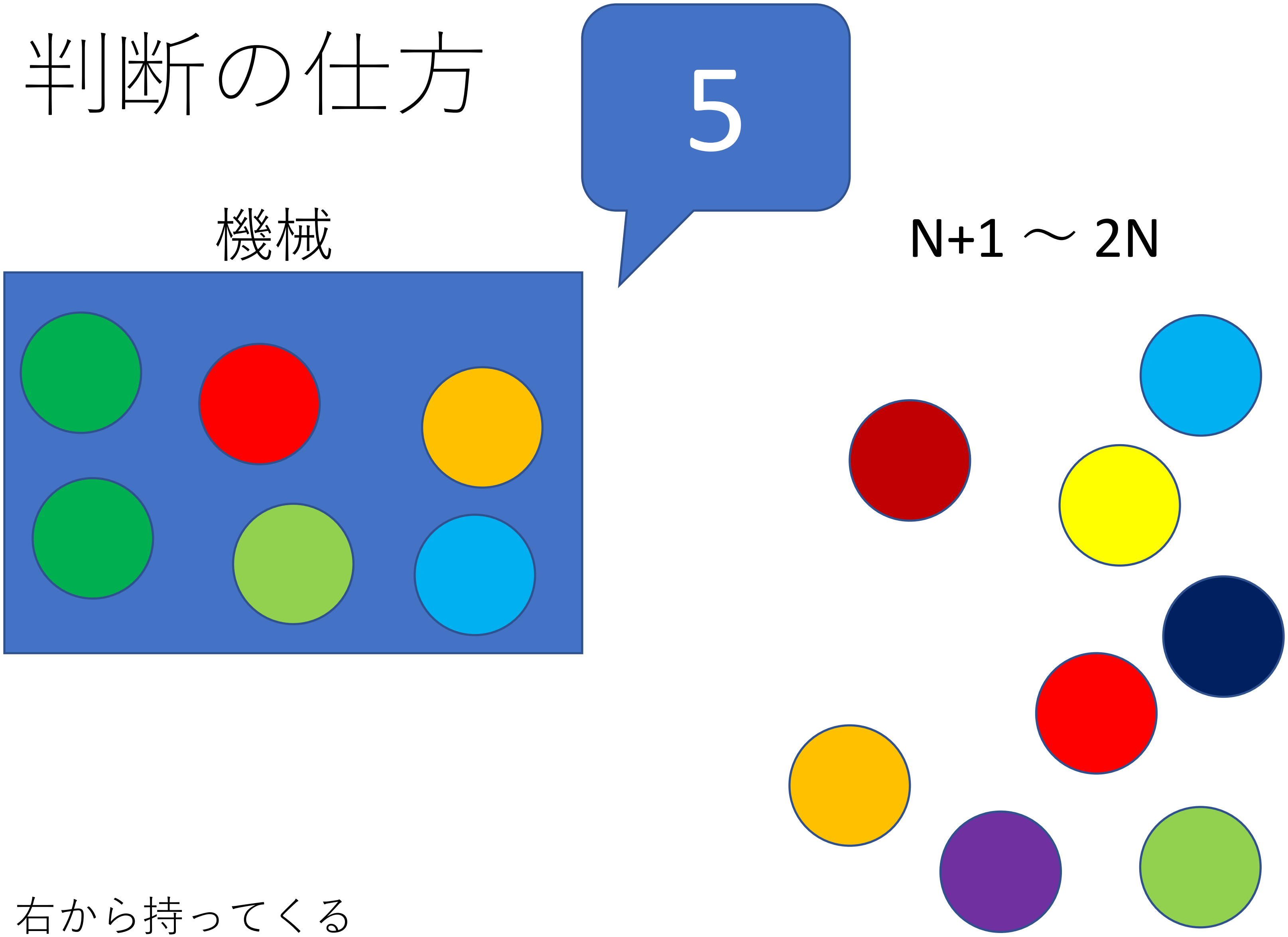
追加しておく

判断の仕方



右から持ってくる
種類数が増えたら別の集合で
ある証拠

判断の仕方

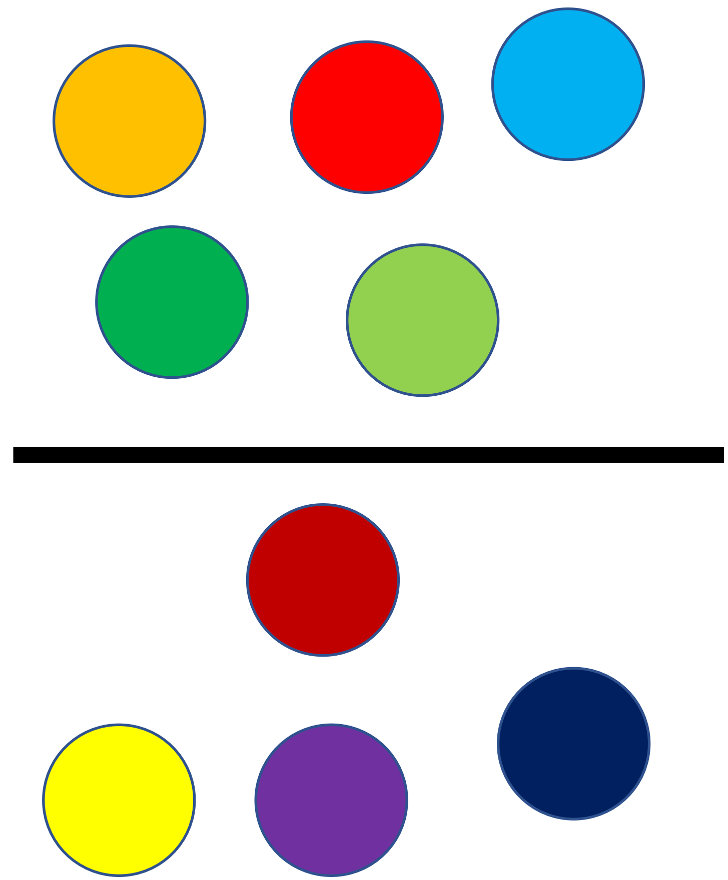


右から持ってくる
種類数が増えなかったら同じ
集合である証拠

判断の仕方

機械

N+1 〜 2N



分割できました
次の人のために機械をきれい
にしておこう

何回クエリを呼ぶか？

- 集合サイズが 1 になるまでに log N ステップかかる
- それぞれに対し、
- 左側の半分を装置に足す
- 右側のそれぞれを足す、取り出す
- 左側の足されたものを取り出す

- 各ステップ 左側合計 N 回、右側合計 2N 回クエリを呼んでいるので、およそ 3 N log N 回

制約

N ≤ 100

N ≤ 15000, 切片1〜N は全部別の種類

N ≤ 15000

N ≤ 38000

N ≤ 39000

N ≤ 40000

N ≤ 41000

N ≤ 42000

N ≤ 43000

グループ分けがない…

- どうしようかな

- 自分で見つけようか

2つの{1,...,N}のグループに分類

- 4N回あれば 2 つに分けられる

- 切片 1 から順に追加
- 追加した後、前の計測結果より増えてたらその種類は未登場
- そうでなければその種類は以前に登場済み
- 未登場と登場済みでそれぞれグループがつくれる

2つの{1,...,N}のグループに分類

- 4N回あれば 2 つに分けられる

- 一つ一つ順に追加するので 2N 回
- 後できれいにするので 2N 回

- この後も考慮すると、合計 $4N + 3N \log N < 70$万

制約

$N \le 100$

$N \le 15000$, 切片1〜N は全部別の種類

$N \le 15000$

$N \le 38000$

$N \le 39000$

$N \le 40000$

$N \le 41000$

$N \le 42000$

$N \le 43000$

方針

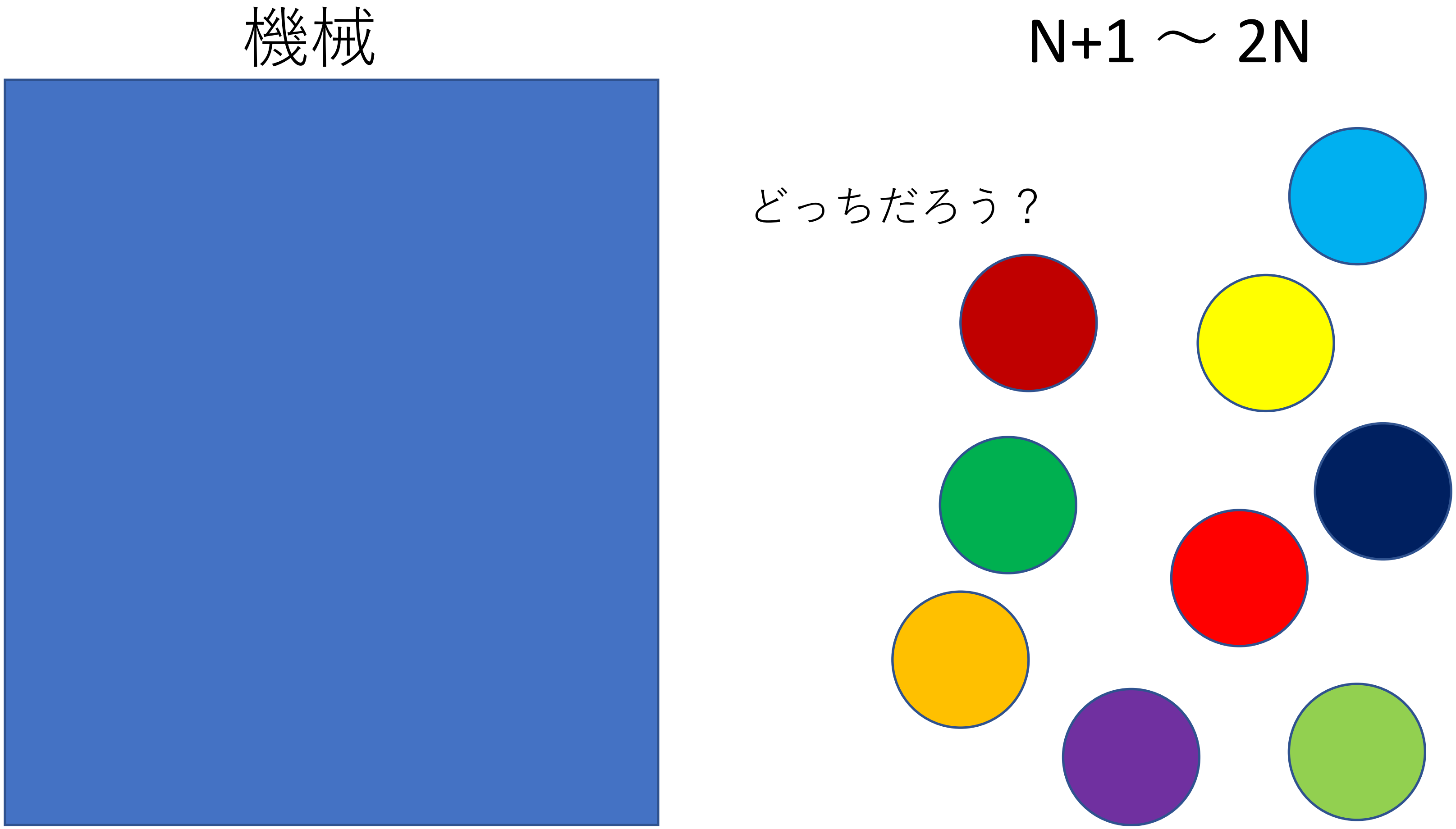
- あとは方針は同じ、ちまちま改善していこう
- Output only ではないけれども Output only 的な傾斜配点だと思って

着眼点 1

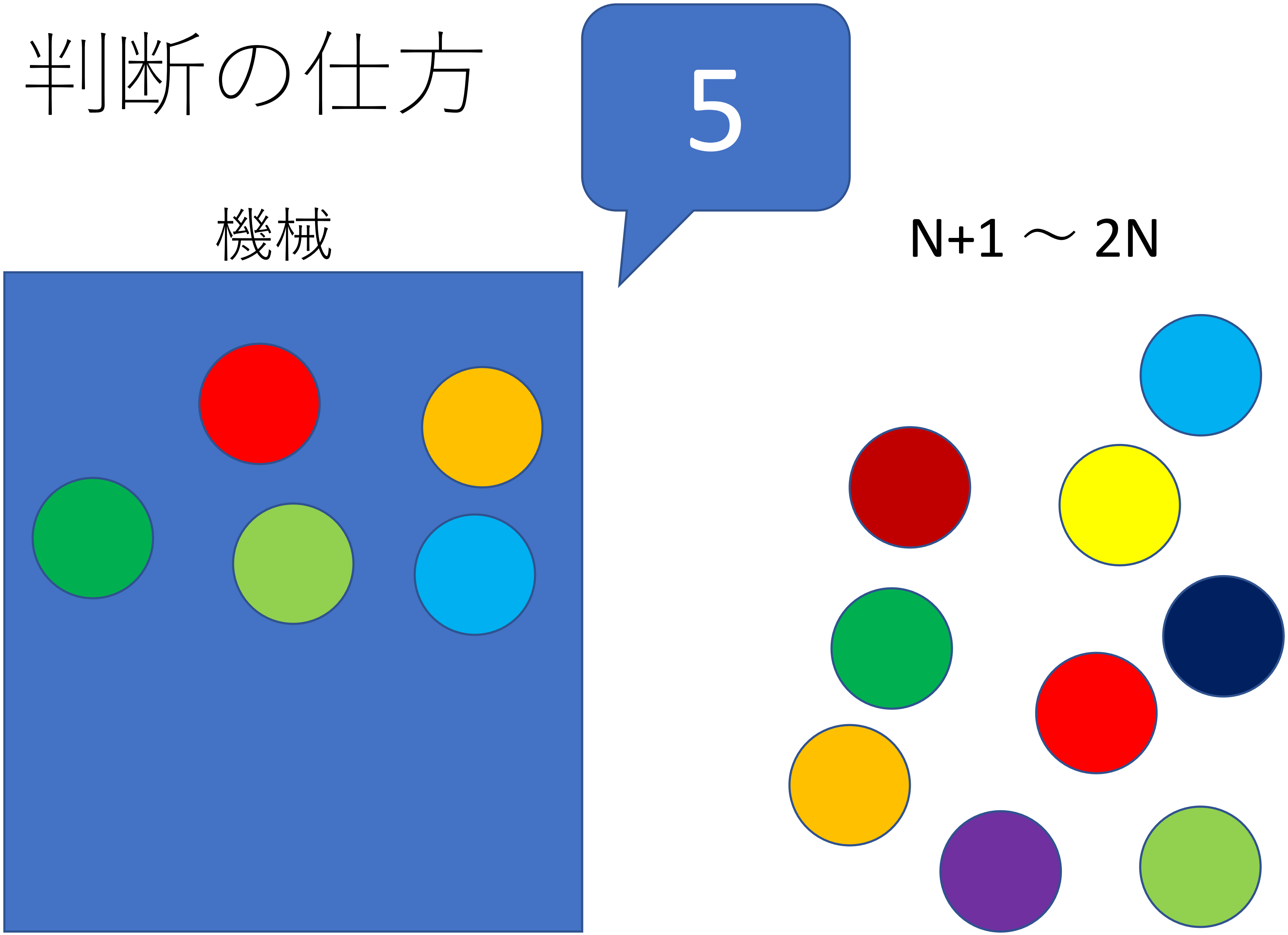
- いちいちきれいにする必要はあるか？

- ないよね
- 分けたい 2 つの分類の外は放置して良いし、分けたいところも in/out を変えるべきものだけ変えればよい
- 最初の 2N -> N*2 にするところも終わった後きれいにする必要はない

判断の仕方

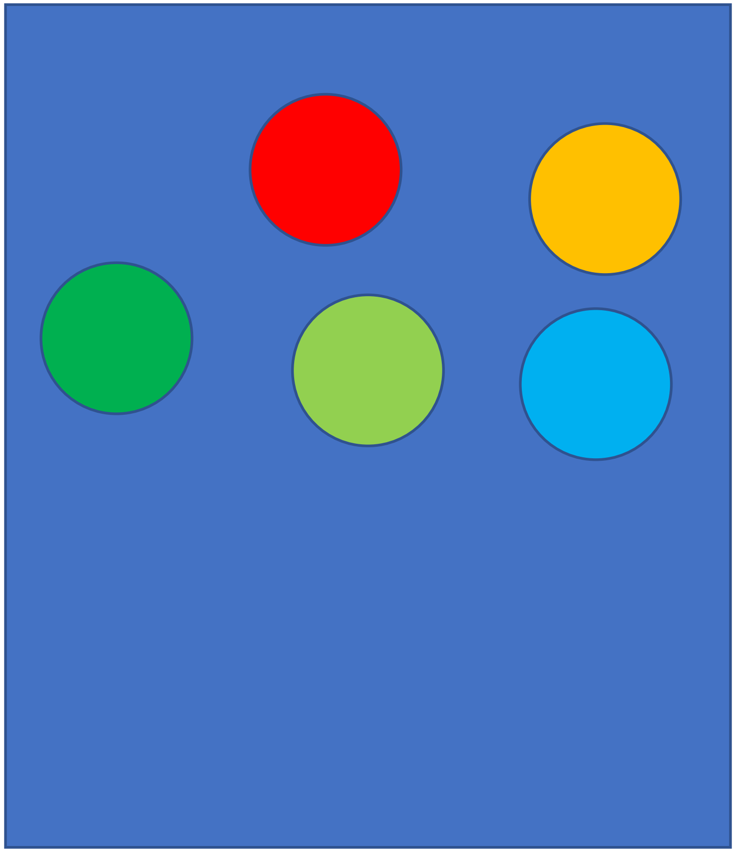


判断の仕方

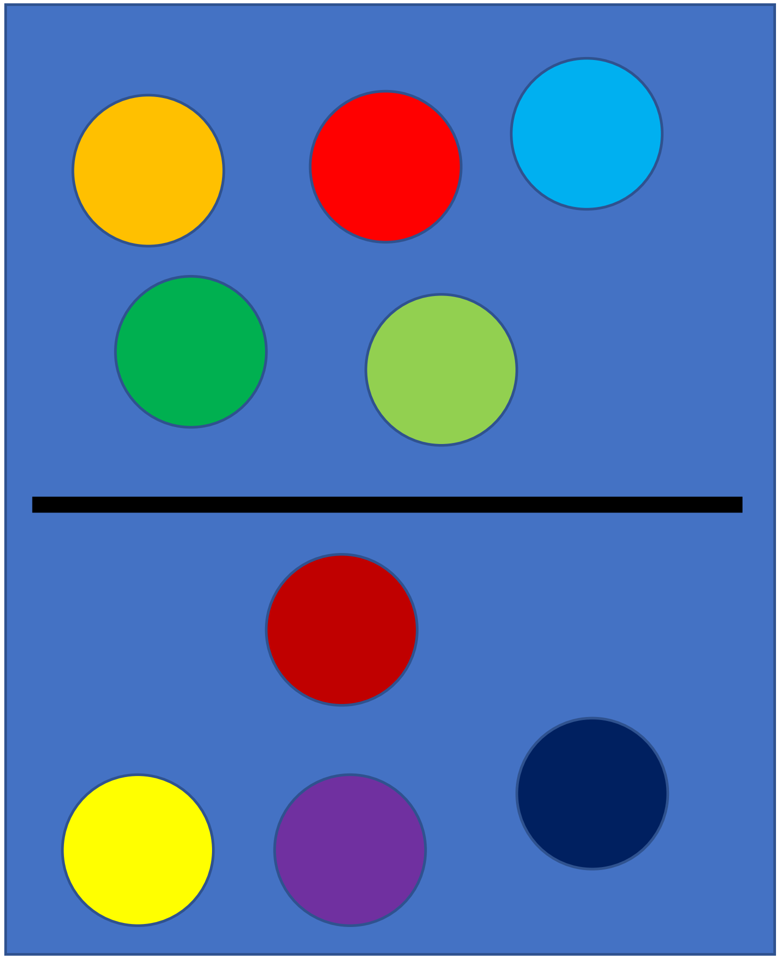


判断の仕方

機械

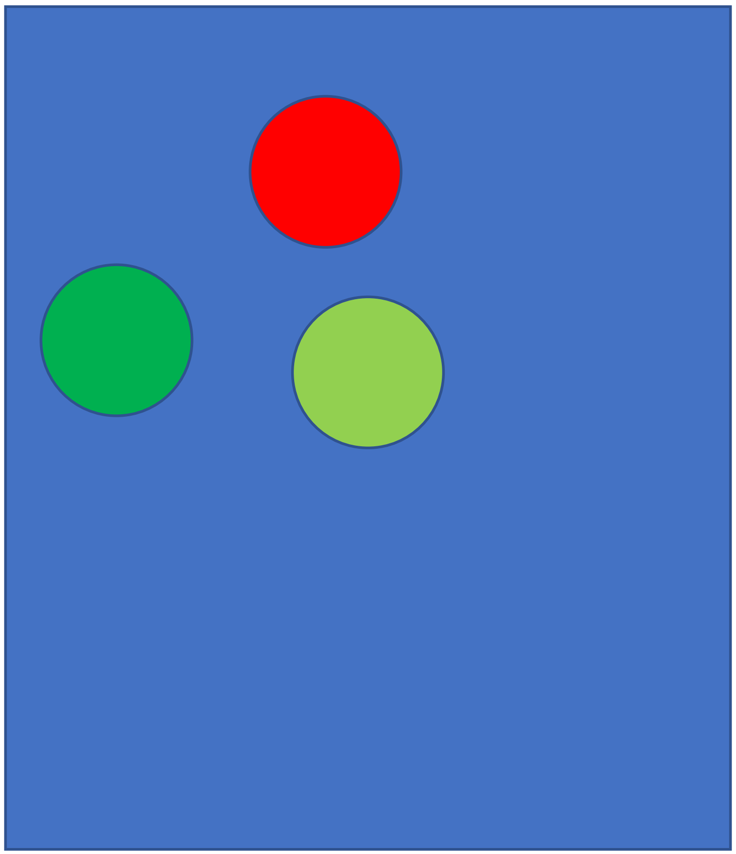


N+1 ～ 2N

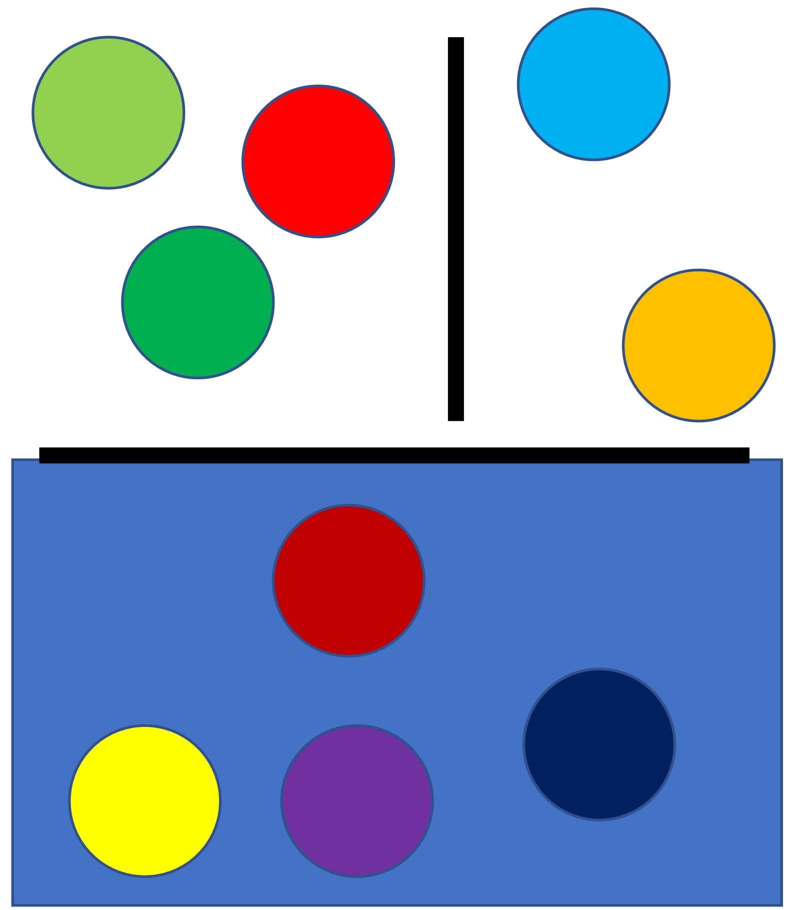


判断の仕方

機械

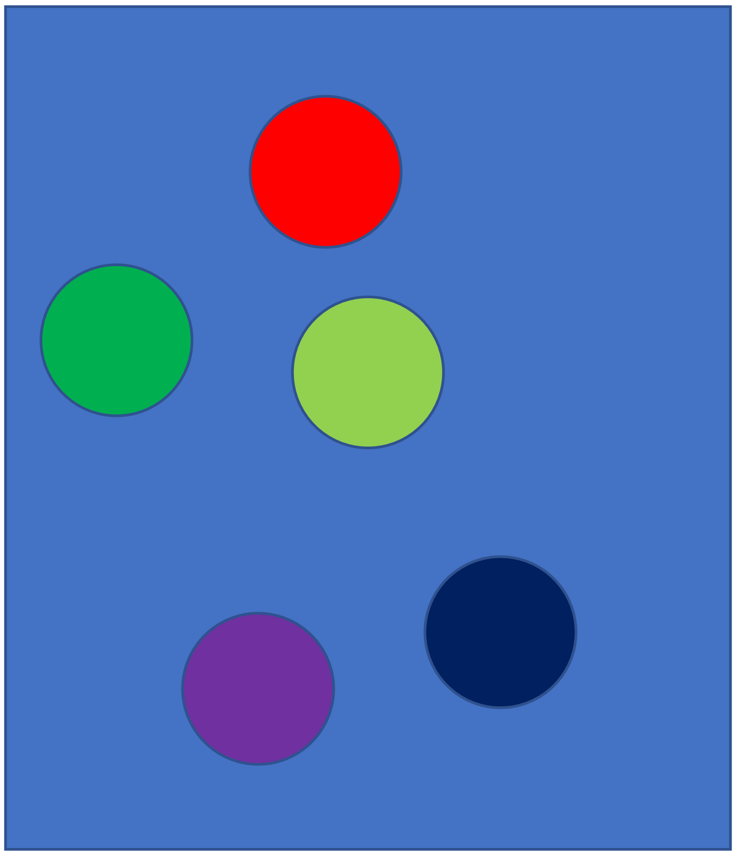


N+1 〜 2N

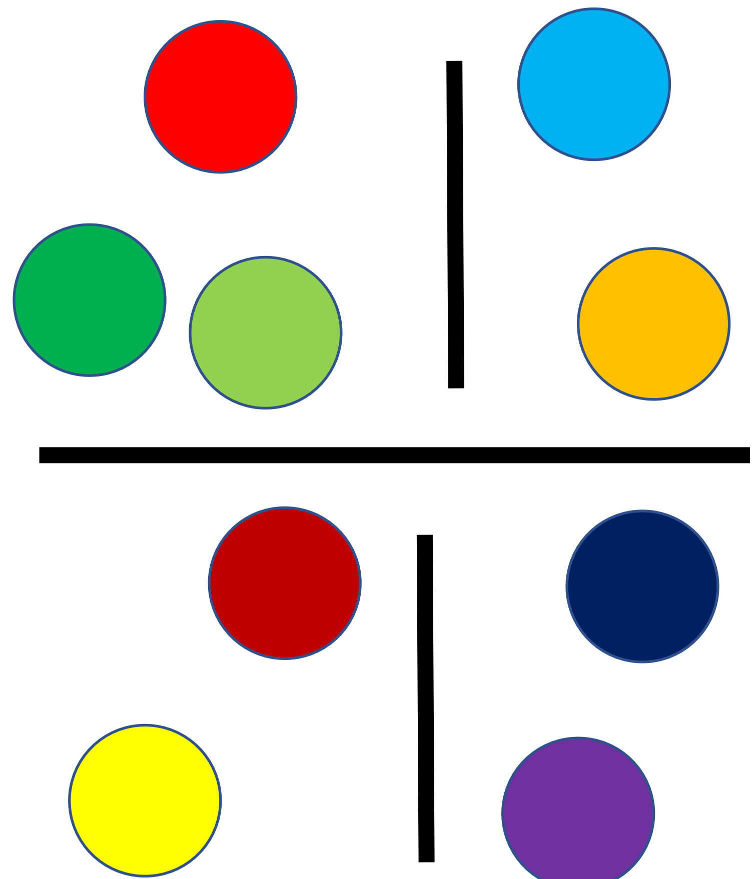


判断の仕方

機械



N+1 ～ 2N



着眼点１

- すると、必要な回数はどのくらいになるか？

- 最初: 2N 回
- 分割統治: log N ステップ
- それぞれ 1～N: 0.5 N 回
- N+1～2N: N 回
- あわせて 2N + 1.5 N log N 回
- ちょっと節約に成功

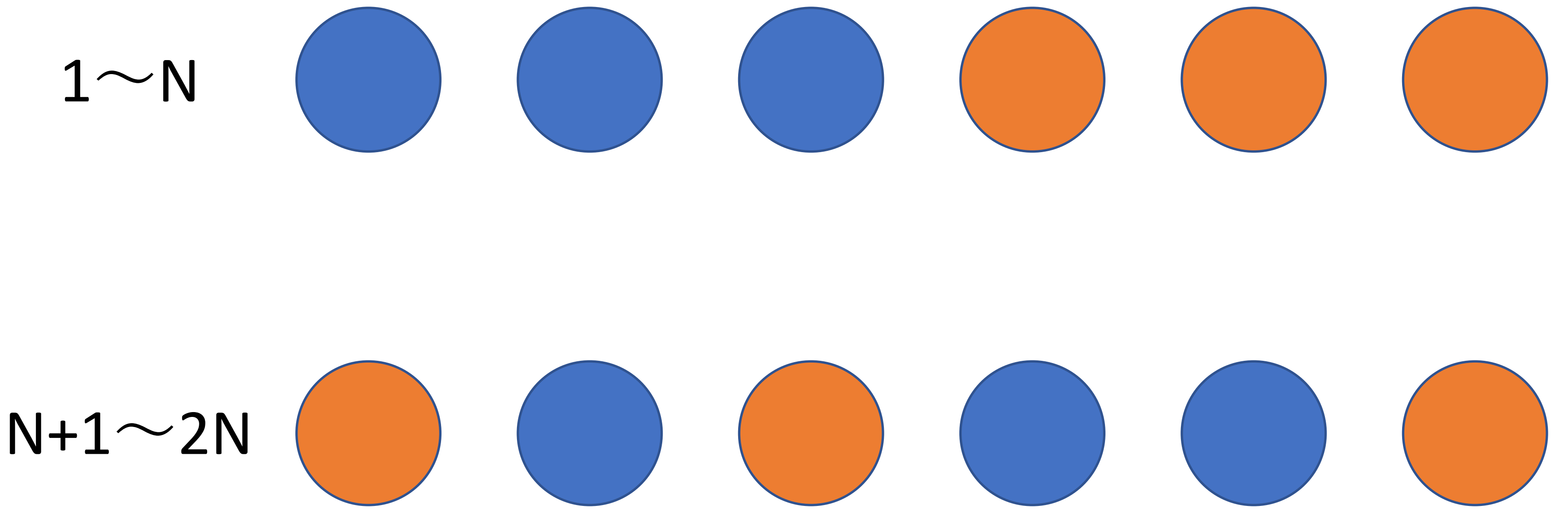
着眼点 2

- 本当にN+1 ～ 2N 全部を出し入れして確定する必要があるか？

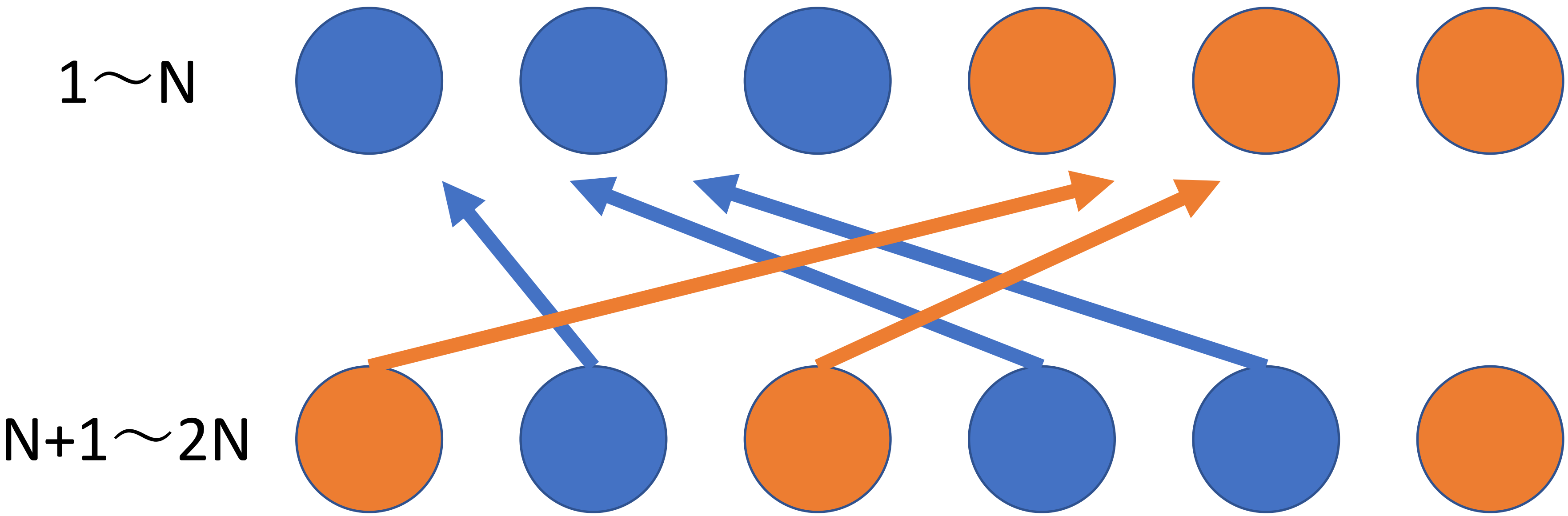
- 実はいらない
- 最後の何個かは確定したりする

- どういうことか？

着眼点 2

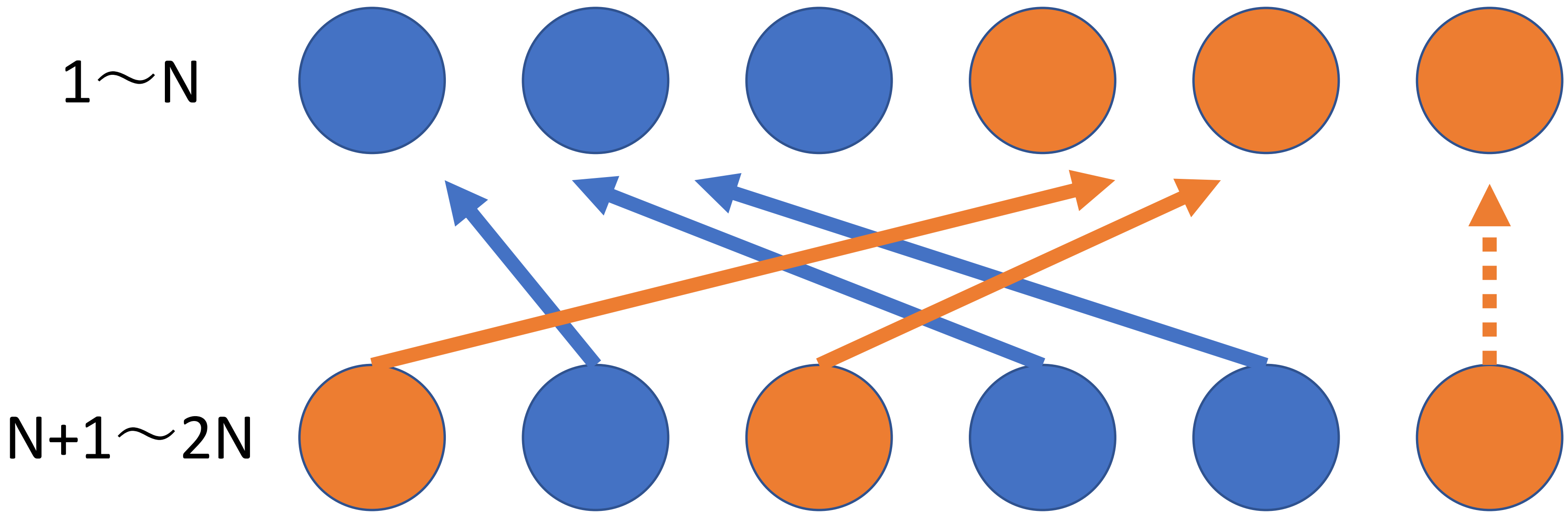


着眼点２



5個真面目に決めてみた、6個目は…

着眼点 2



5個真面目に決めてみた、6個目は…
どうみても右側ですね…

着眼点 2

- ということで、各分割統治の再帰ステップのうち、最後の 1 個は追加すらする必要がないっぽい (運が良ければ 2個以上いらないかも)
- 再帰関数の呼ばれる回数は 2N 回くらい
- これで最低 2N 回減ったな (やったぜ。)

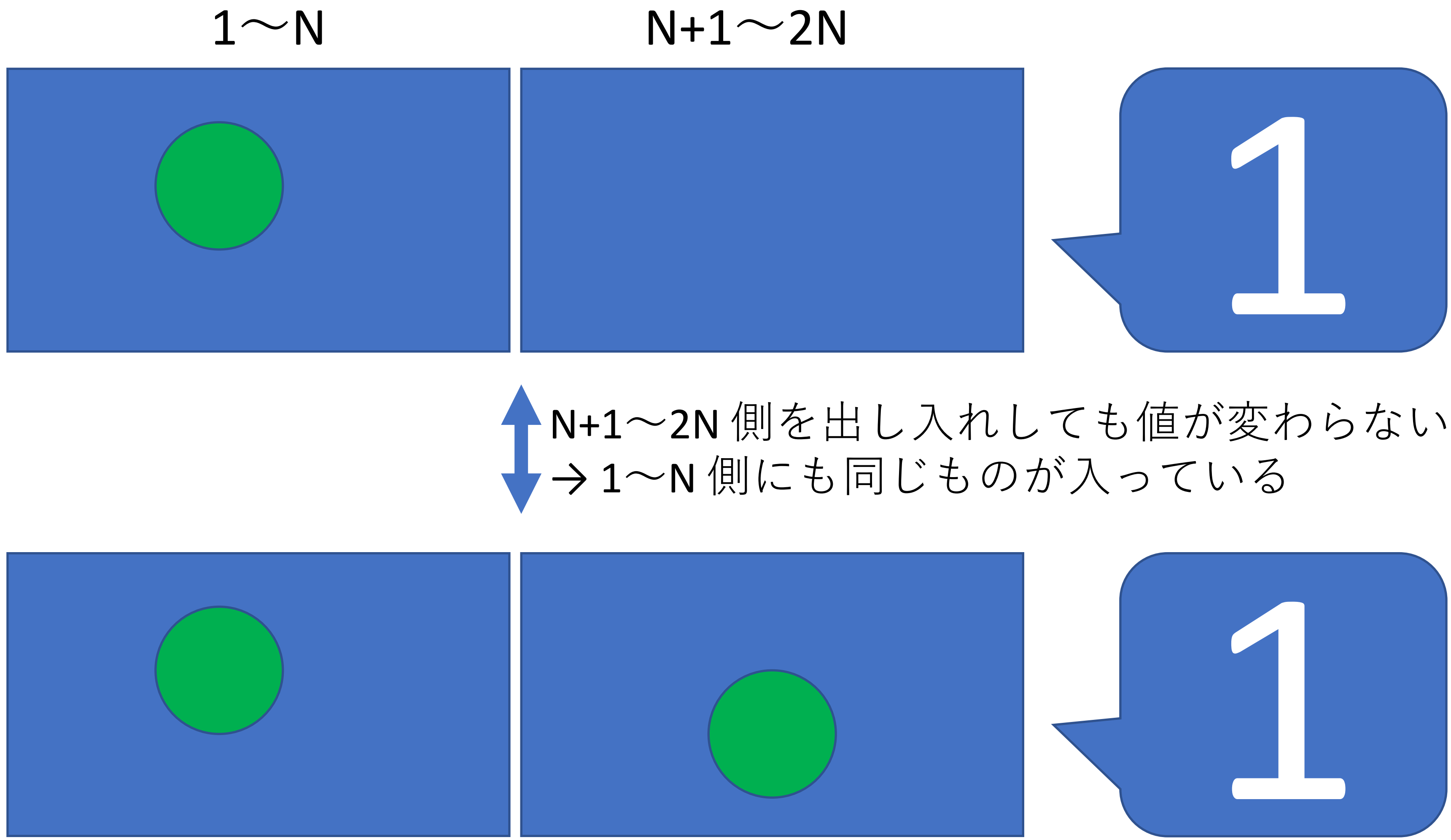
着眼点 2

- ちょっと待った
- 最後の 1 個が切り替わらないと、次そこを切り替える時増えるのか減るのかわからなくなりませんか？

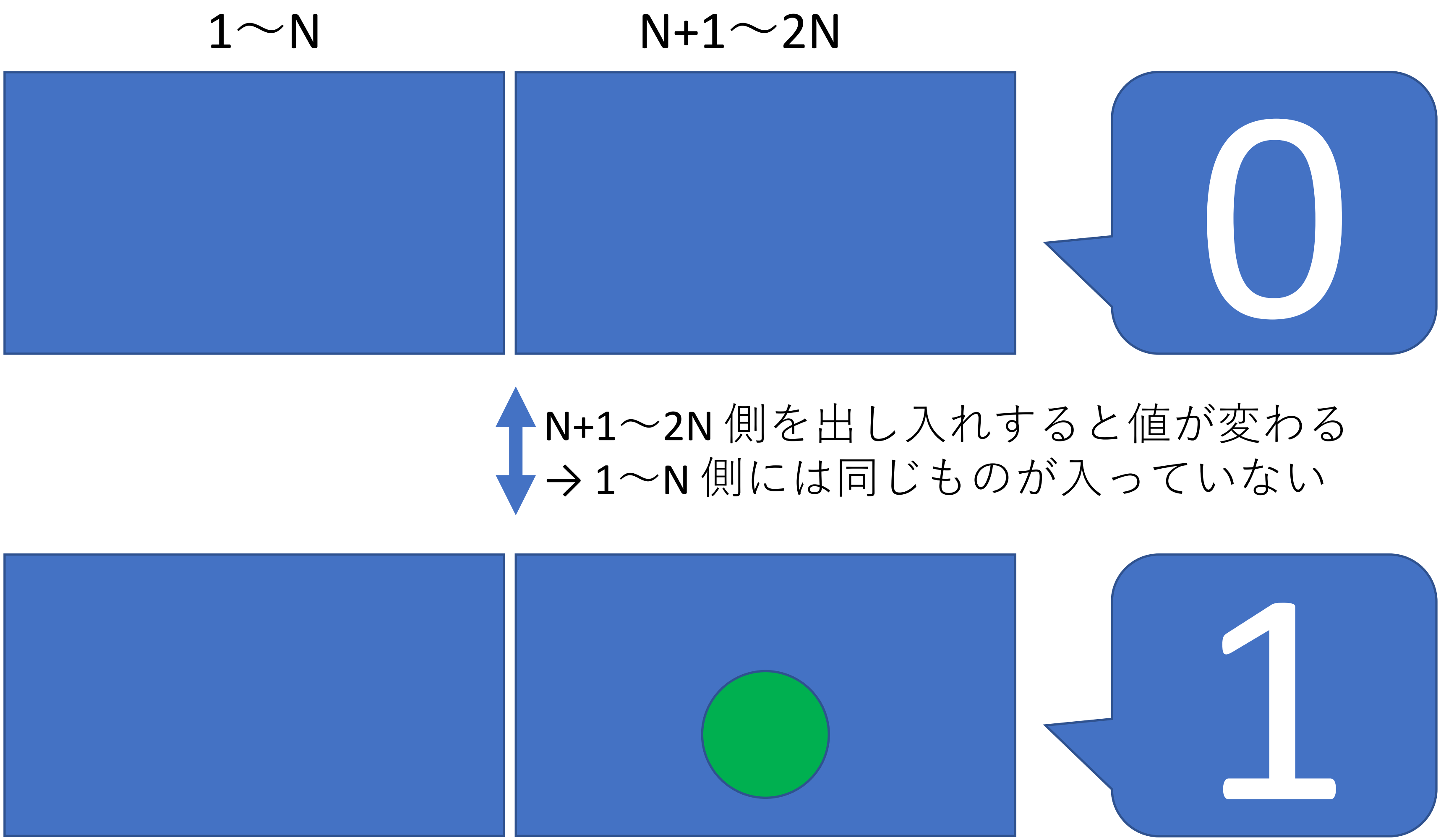
着眼点2

- そもそもクエリの答えの増加と減少に何か意味はあるのか？

着眼点2



着眼点2



着眼点 2

- そもそもクエリの答えの増加と減少に何か意味はあるのか？
- 増加か減少か？ではなく、直前と変化したかどうかに意味があった。

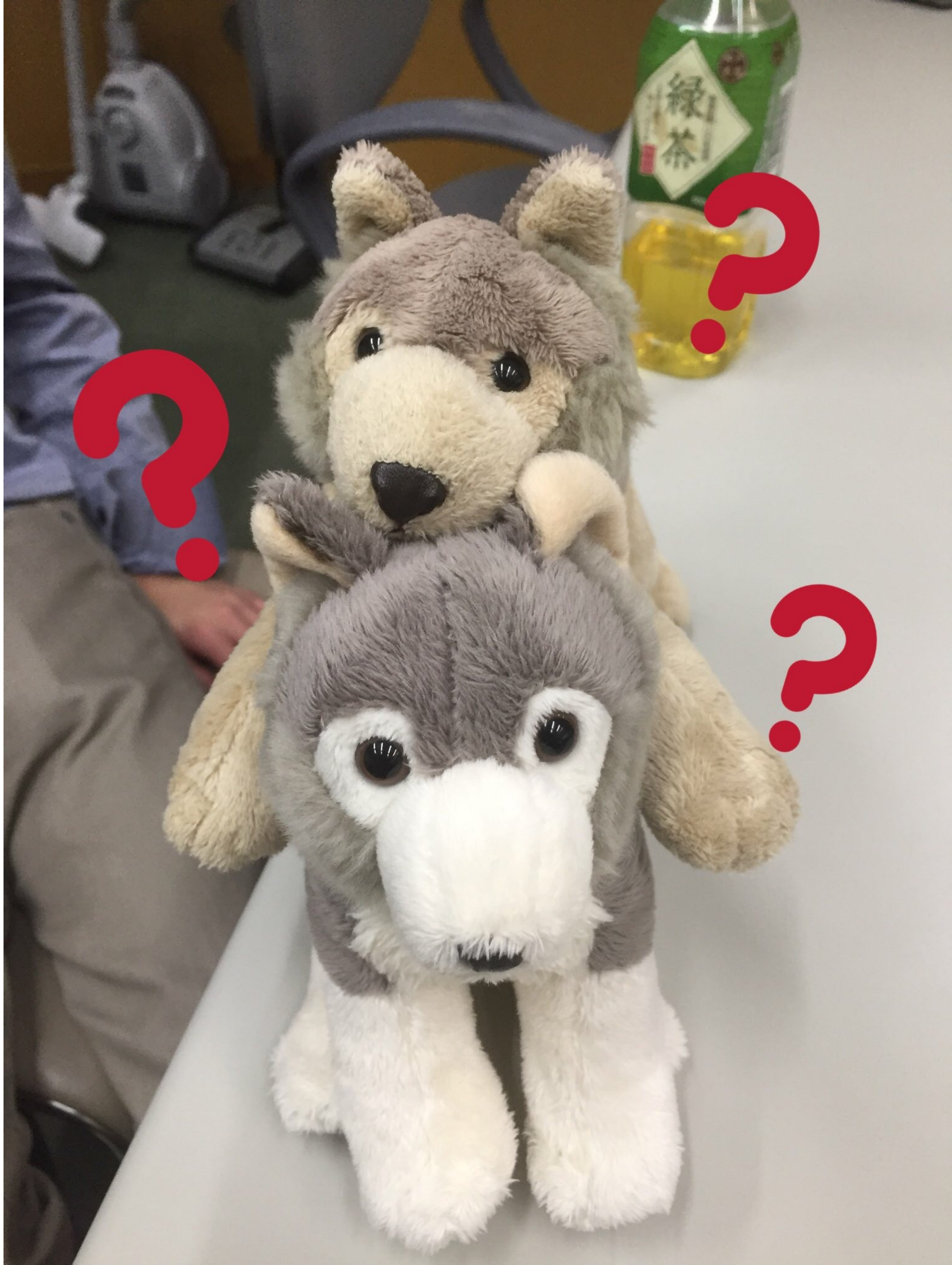
- じゃあ 1 個余ったものもそのまま放置しておいて問題ないね

着眼点 2: 運を良くしたい

- ~~パワーストーンを買いましょう~~
- 最悪ケース(1つも1 個しかスキップできない)を回避するには、2つにグループ分けをしたあと random_shuffle するのが良い

Random Shuffle ってどうやるの

- swap(a[rand()%N], a[rand()%N]) を繰り返して擬似ランダムシャッフルをしていて楽しいか？



Random Shuffle

- random_shuffle(a,a+N); を
  - たぶん <algorithm> にある
- for(int i=0;i<N;i++)swap(i,ra
- いずれにせよ、 O(N) でランダムな順列の一つを等確率で作れて最高

着眼点3

- 今までは何も言ってなかったが、ステップ数・回数の計算では暗黙の了解で半分ずつに分けていた
- 本当にこれが最適なのか？

着眼点 3

- 分割統治パートを定式化しよう

- 1. 1～N 側の候補を p:1-pに分ける
- 2. p 側を全て反転
- 3. N+1～2N 側を全て出し入れして、どちら側にあるかを決める
- 4. 再帰

着眼点 3

- N 種類を割り当てるのにかかるクエリ回数を f(N) = tN log N とすると、

$$
\begin{aligned}
& f(N) = f(pN) + f\left((1-p)N\right) + (1+p)N \\
& \iff tN\log N = tpN\log(pN) + t(1-p)N\log(pN) + (1+p)N \\
& \iff 0 = tpN\log p + t(1-p)N\log(1-p) + (1+p)N \\
& \iff t = \frac{-1-p}{p\log p + (1-p)\log(1-p)}
\end{aligned}
$$

- t が最小となる p ってどこだろう？

着眼点 3

- 微分をしましょう

$$\left(\frac{-1-p}{p\log p+(1-p)\log(1-p)}\right)'=0$$

$$\iff p\log p+(1-p)\log(1-p)-(1+p)(\log p-\log(1-p))=0$$

$$\iff 2\log(1-p)-\log p=0$$

$$\iff \frac{(1-p)^2}{p}=1$$

$$\iff p=p^2-3p+1$$

$$\boxed{\iff p=\frac{3-\sqrt{5}}{2}\simeq 0.382}$$

着眼点 3

- ということで、 p=0.38 くらいのところで分けて小さい方を反転すれば回数が最適になることがわかる。

- ちなみに微分しなくても、DP で最小回数を達成するための分割方法を求めることができる
- 必要な回数は約 1.44 N log N になる

まとめ

- 着眼点 1: 一回一回きれいにせず、装置に入れるべきものは置きっぱなしにする
- 着眼点 2: 各再帰呼び出しで最後の1個は見なくてよい
  - 2α: random_shuffle すると減る (なくてもよい)
- 着眼点 3: 半分ではなく、0.38 くらいで分割

- 全部合わせると最悪ケースは何回？
- N = 43000 だと… 999976回！