

An underwater scene featuring a large school of small, dark fish swimming in clear blue water. A larger, flat, dark fish, likely a stingray, is visible in the lower-left quadrant. The background shows the sunlit surface of the water with ripples and light filtering through.

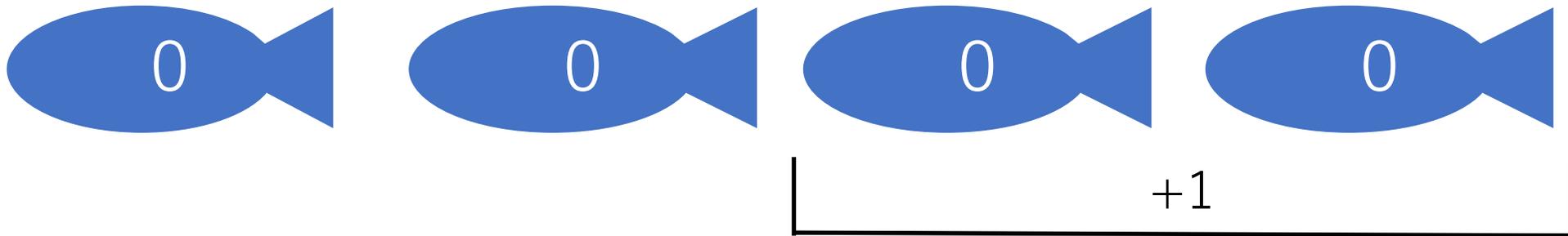
FISH 3

Editorial : blackyuki

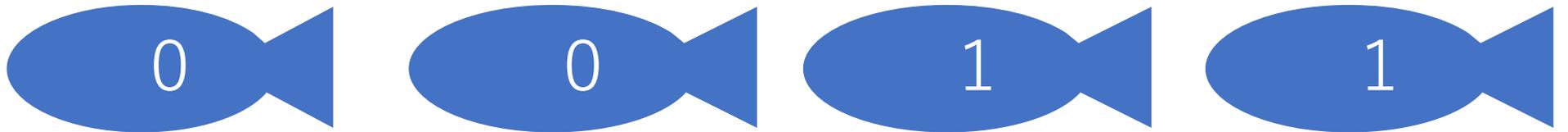
問題概要

- N 匹の魚のIQをそれぞれ決まった値まで増やしたい
- 操作A：魚 k のIQを D だけ増やす
- 操作B：魚 $k, k + 1, \dots, N$ のIQを1ずつ増やす
- 操作Aの回数の最小値は？
- Q 個の区間クエリに答える

入力例 1



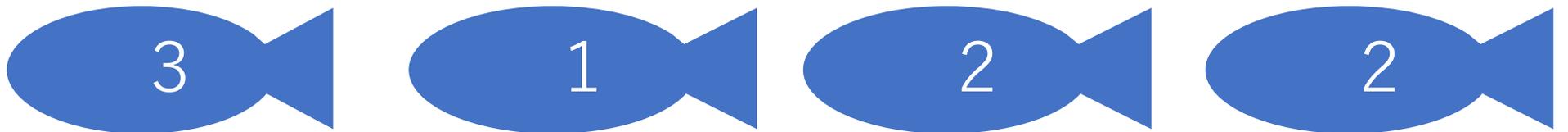
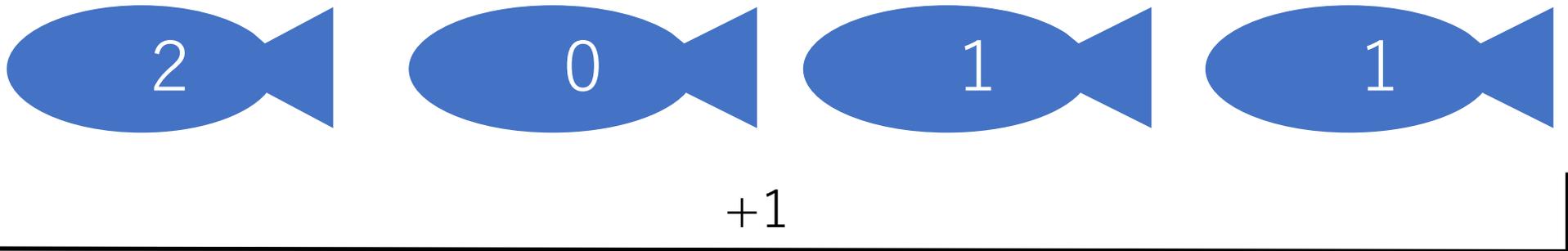
D=2



C=[3,1,2,1]

↓ +2

(L,R) = (1,3)



問題の性質

クエリ

最適化

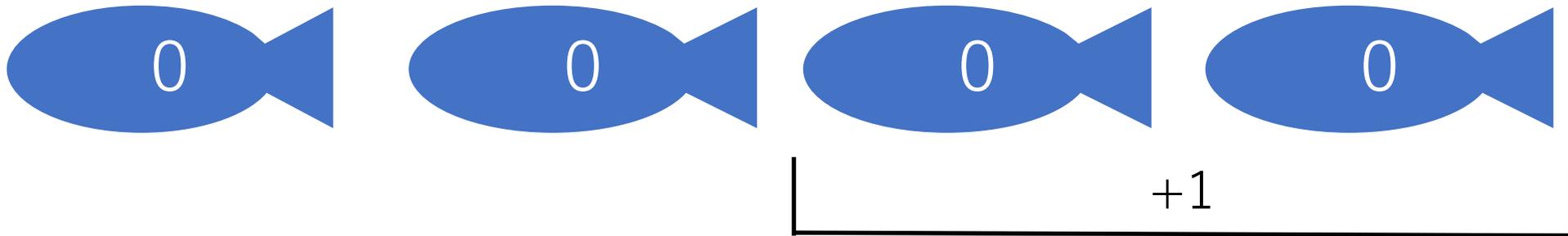
小課題 1

$O(NQ)$

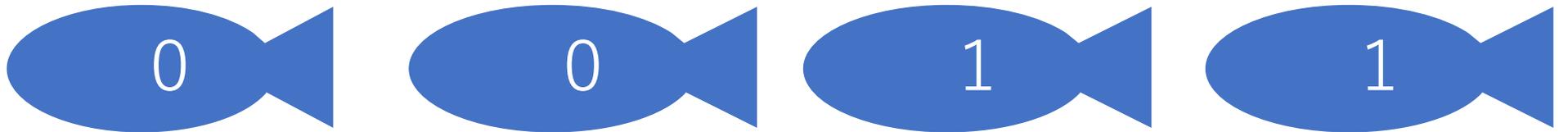
気づき 1

- クエリの区間外の魚は無視してよい

入力例 1



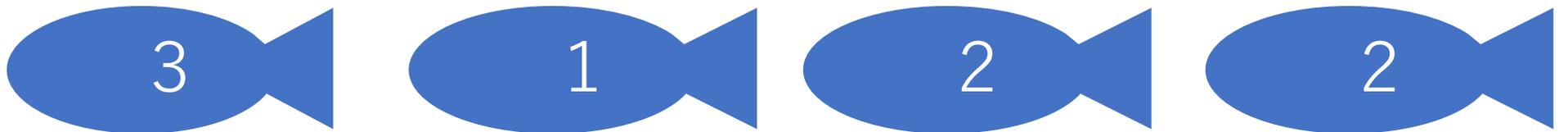
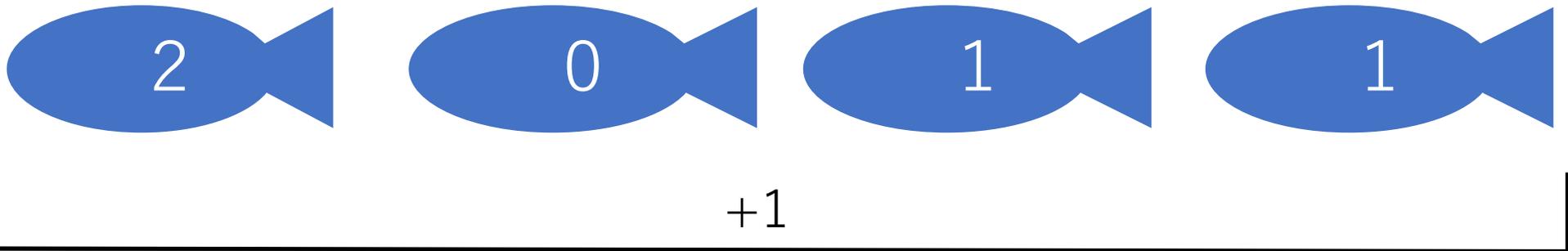
D=2



C=[3,1,2,1]

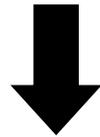
↓ +2

(L,R) = (1,3)



気づき 1

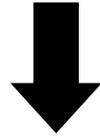
- クエリの区間外の魚は無視してよい



- $Q=1, (L,R)=(1,N)$ の場合が $O(N)$ で解ければ、
全体 $O(NQ)$ で解ける

問題の緩和

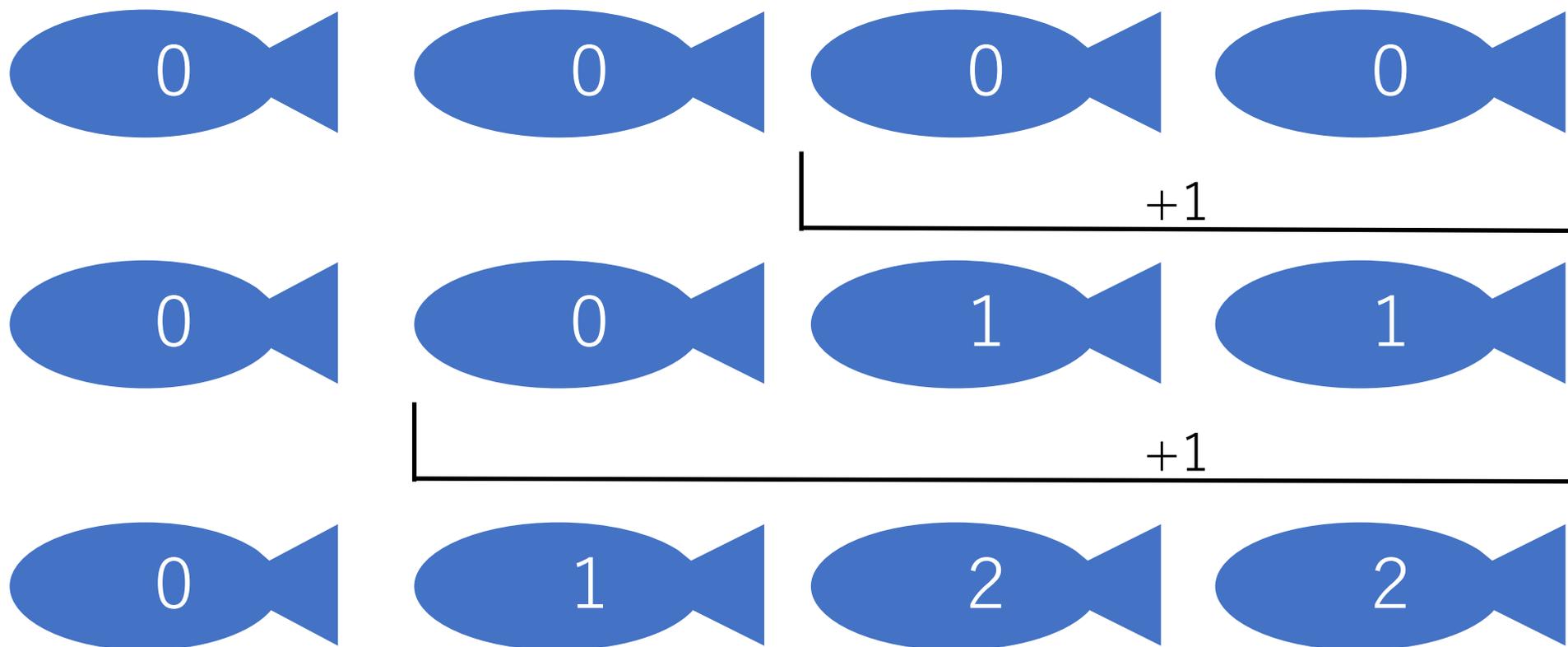
- 操作Aの回数の最小値を求めよ



- 操作Aの回数の最小値が0回かどうか判定せよ

問題の緩和

- 操作Bのみで作れる魚のIQ列を考える



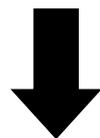
問題の緩和

- 操作Bのみで作れる魚のIQ列
 - • • 任意の広義単調増加列

(証明の概略：差分をとる)

問題の緩和

- 操作Aの回数の最小値が0回かどうか判定せよ

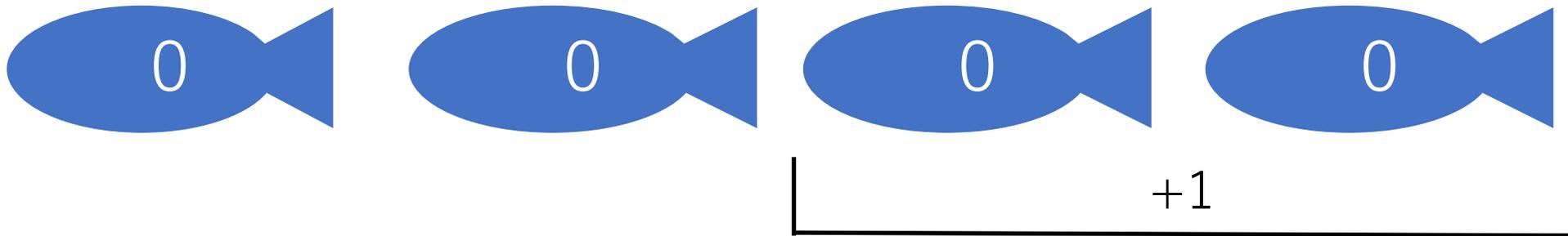


- 数列が広義単調増加か判定すればよい

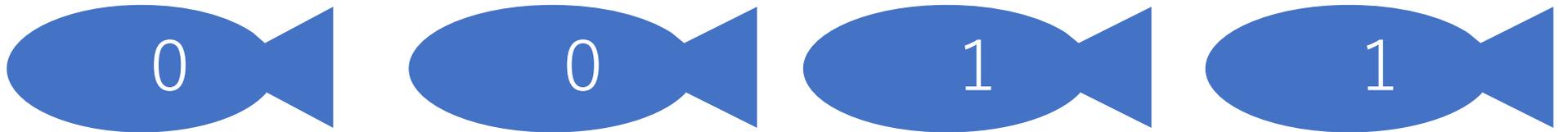
気づき 2

- 操作の順番は関係ない
- 必要な情報はどの魚にどの餌を何回あげたか

入力例 1



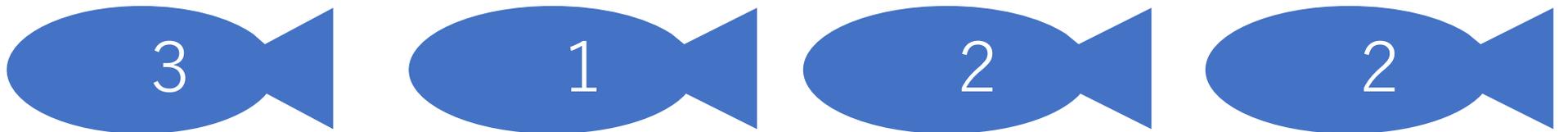
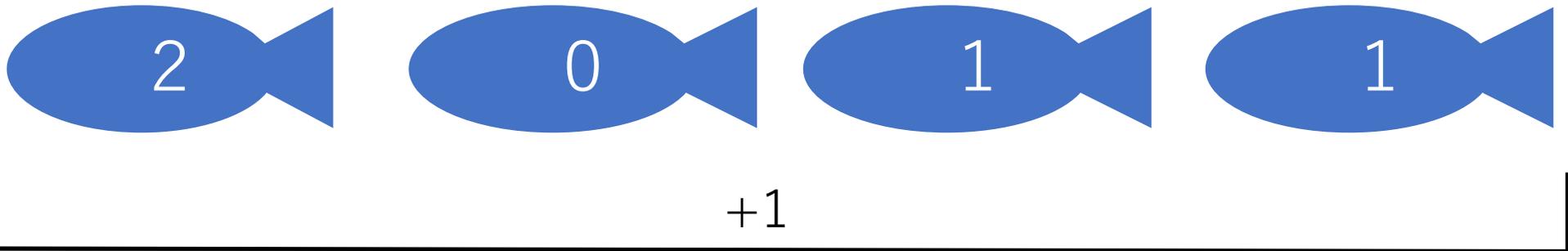
D=2



C=[3,1,2,1]

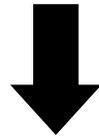
↓ +2

(L,R) = (1,3)



気づき 2

- 操作の順番は関係ない
- 必要な情報はどの魚にどの餌を何回あげたか

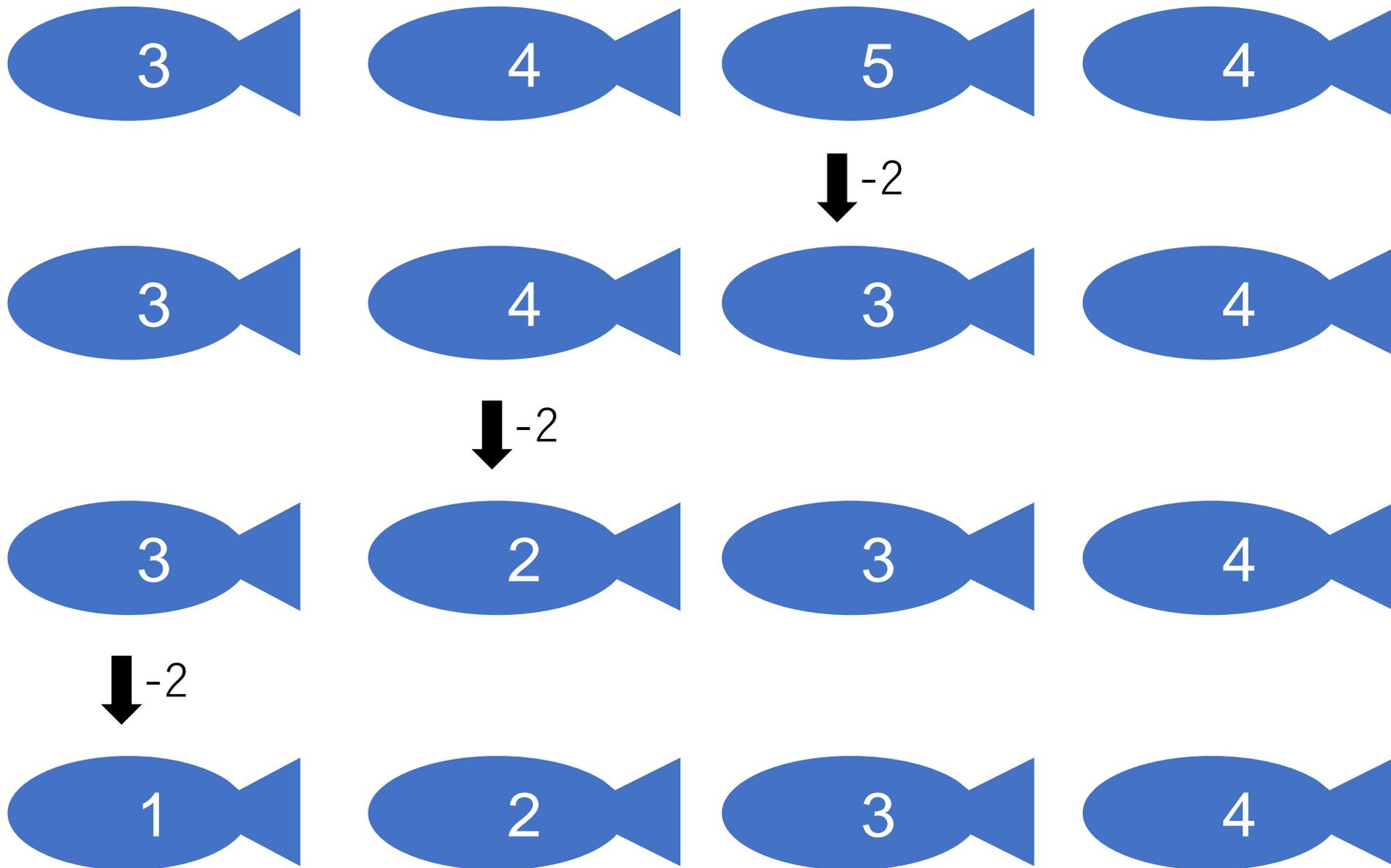


- 例えば初めに操作Bをまとめて行ってから操作Aをまとめて行う、としてよい

気づき 3

- 操作を逆順に見た方が分かりやすい
- 最初魚のIQは C_i
- ある魚を選んでD減らす、という操作をできるだけ少ない回数行って広義単調増加列にしたい

D=2



気づき 4

- 右の魚から貪欲にIQを削っていけばよい
- 左端の魚のIQが負になれば不可能

小課題 1 が通る (9点)

```
ll sol(ll n, ll d, vi v){
    ll ans=0, n=v.size();
    for(ll i=n-2; i>=0; i--){
        if(v[i]>v[i+1]){
            ll k=(v[i]-v[i+1]+d-1)/d;
            ans+=k;
            v[i]-=k*d;
        }
    }
    if(v[0]<0) return -1;
    return ans;
}
```

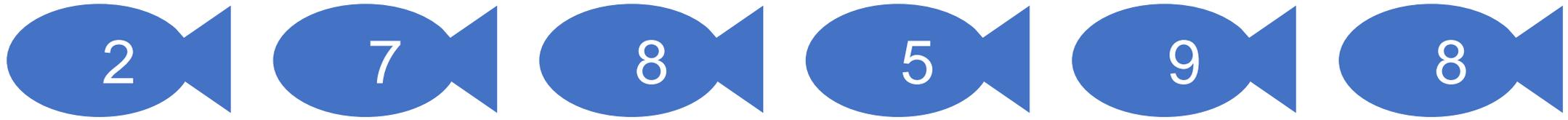
小課題 2

$$C_i \leq 1$$

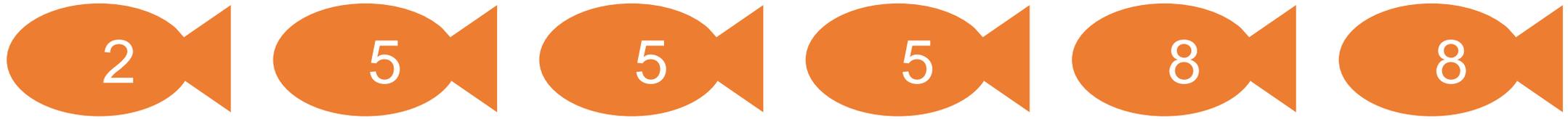
累積和と二分探索で頑張る

小課題 3

$D=1$



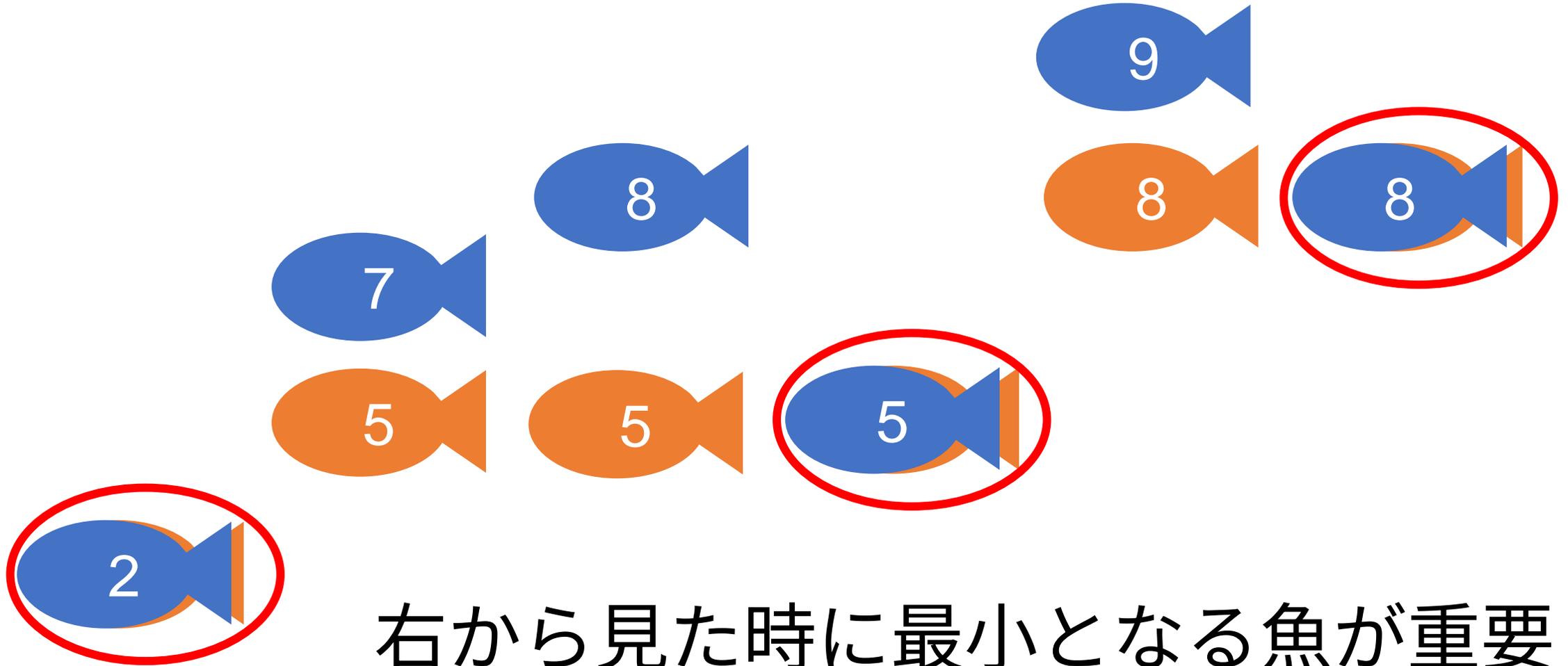
↓ D=1



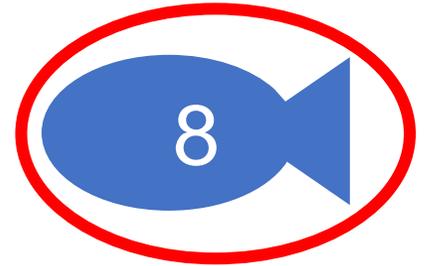
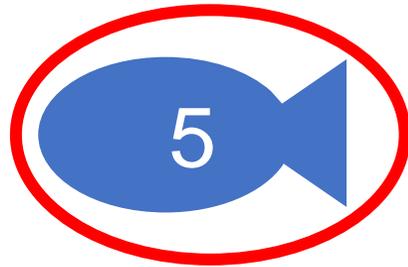
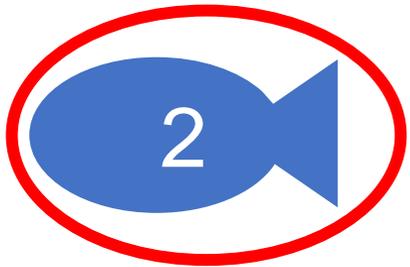
始状態と終状態の総和の差から操作回数が計算できる

→終状態を求めたい

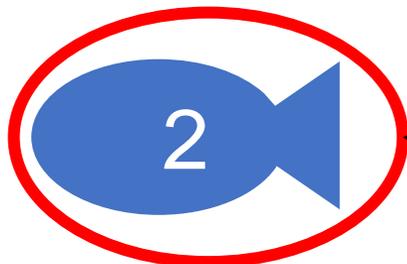
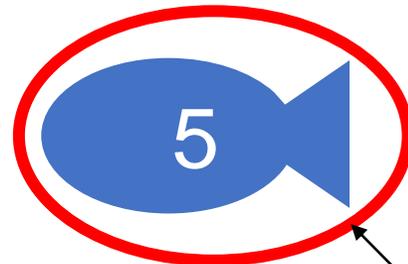
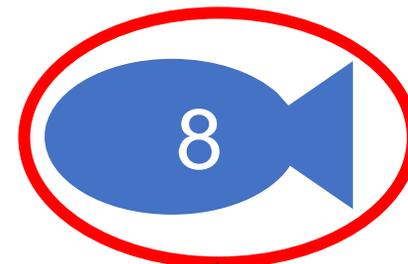
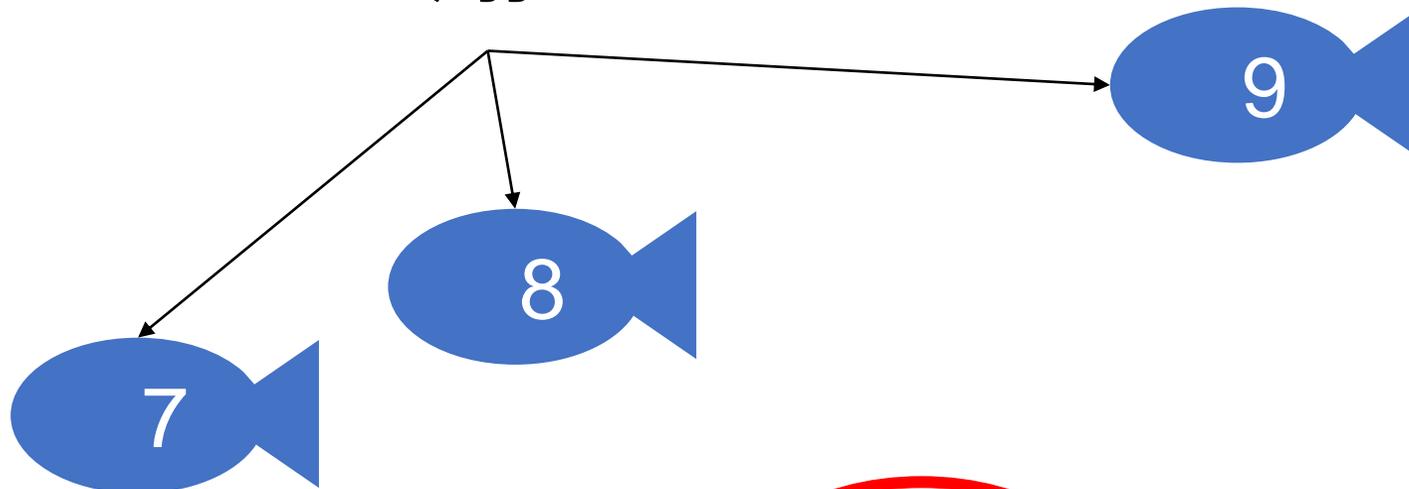
2次元平面にプロットします



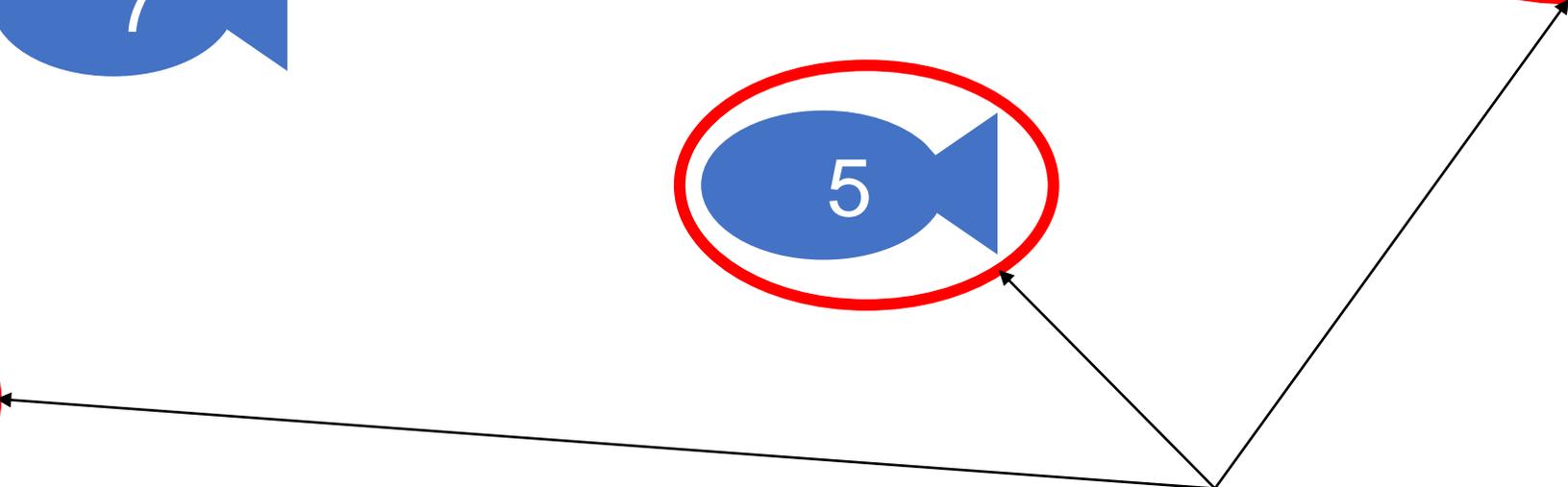
右から見た時に最小となる魚が重要



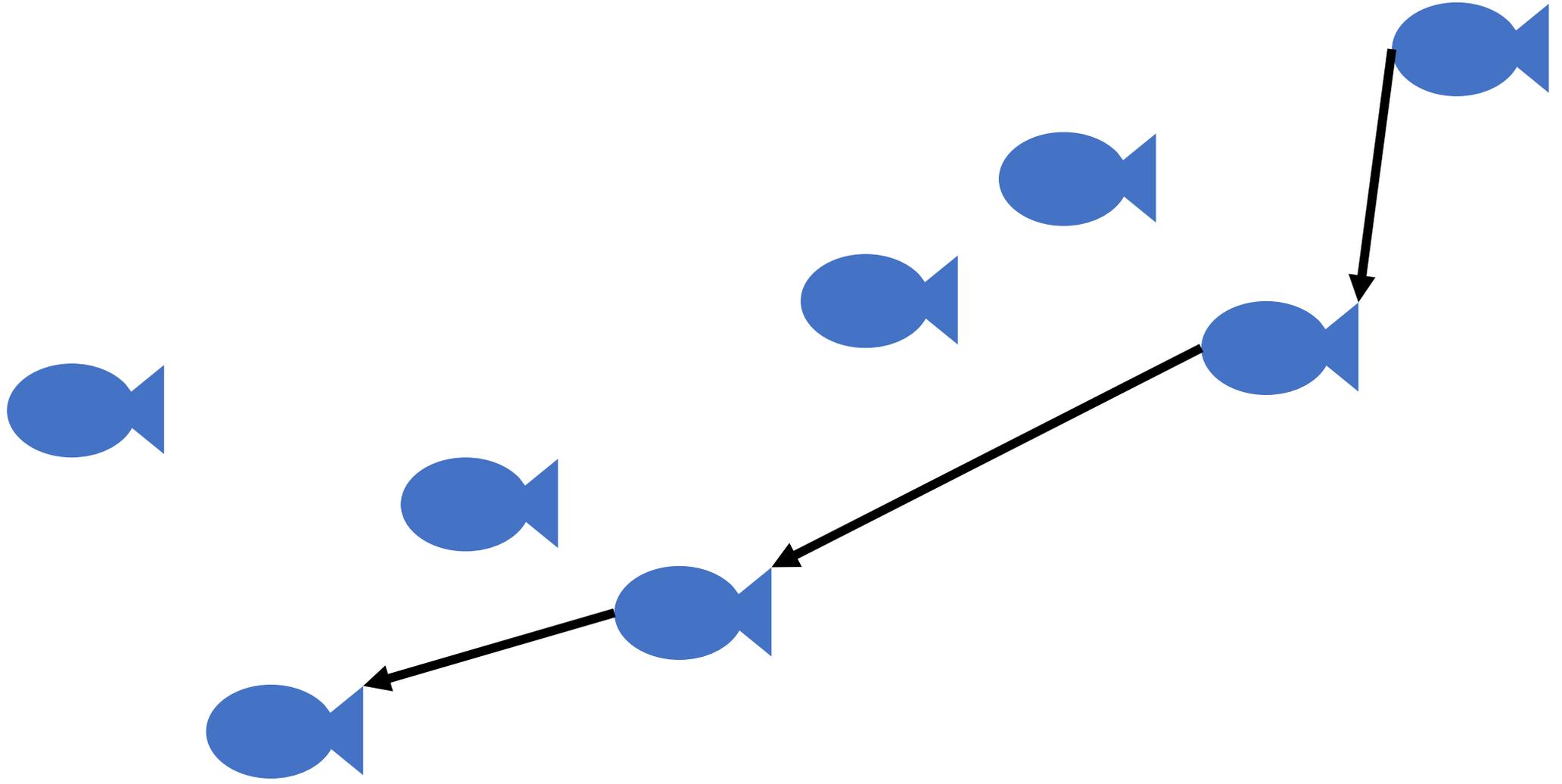
鰭



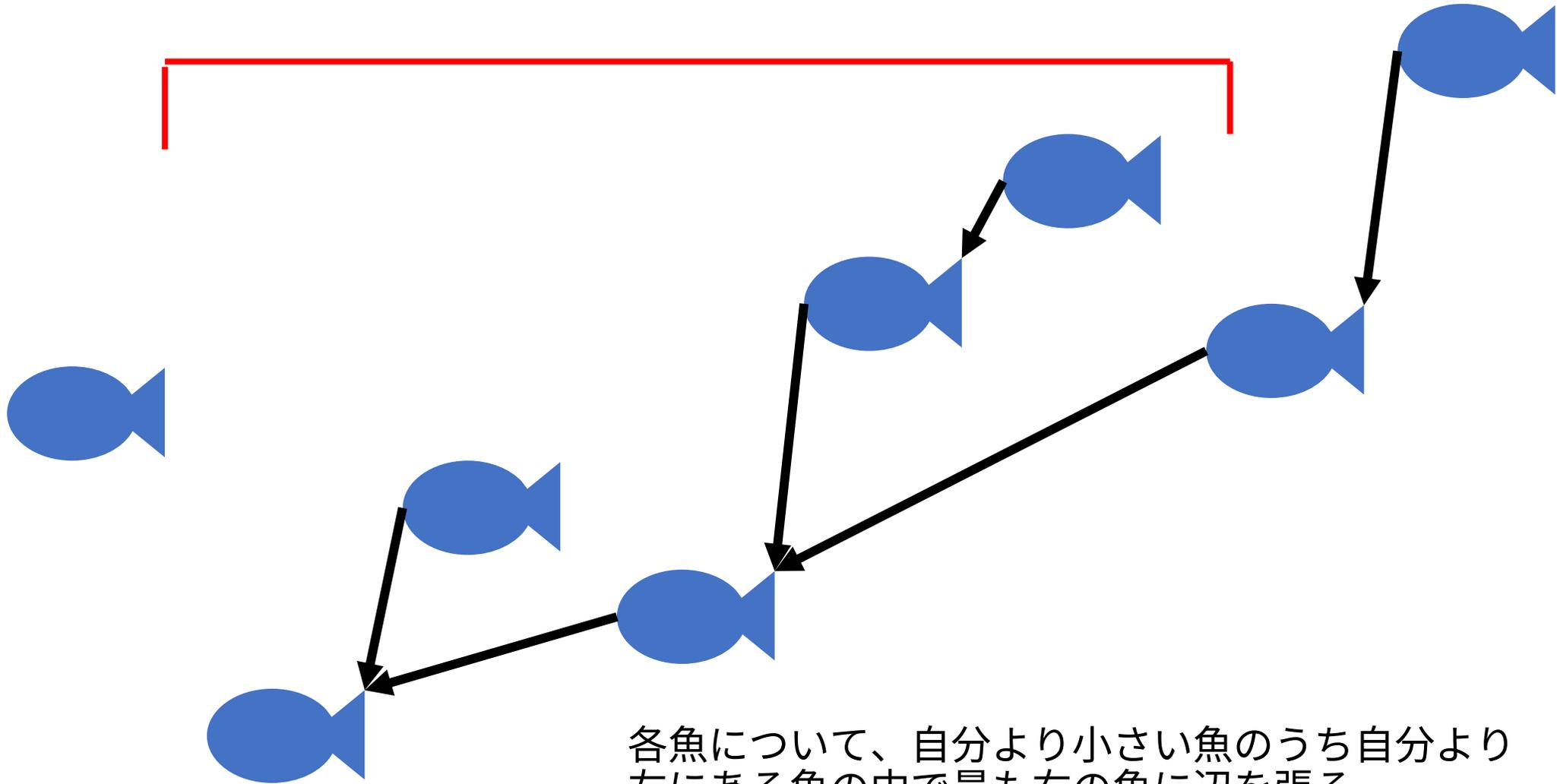
秋刀魚



秋刀魚だけを取り出したい！

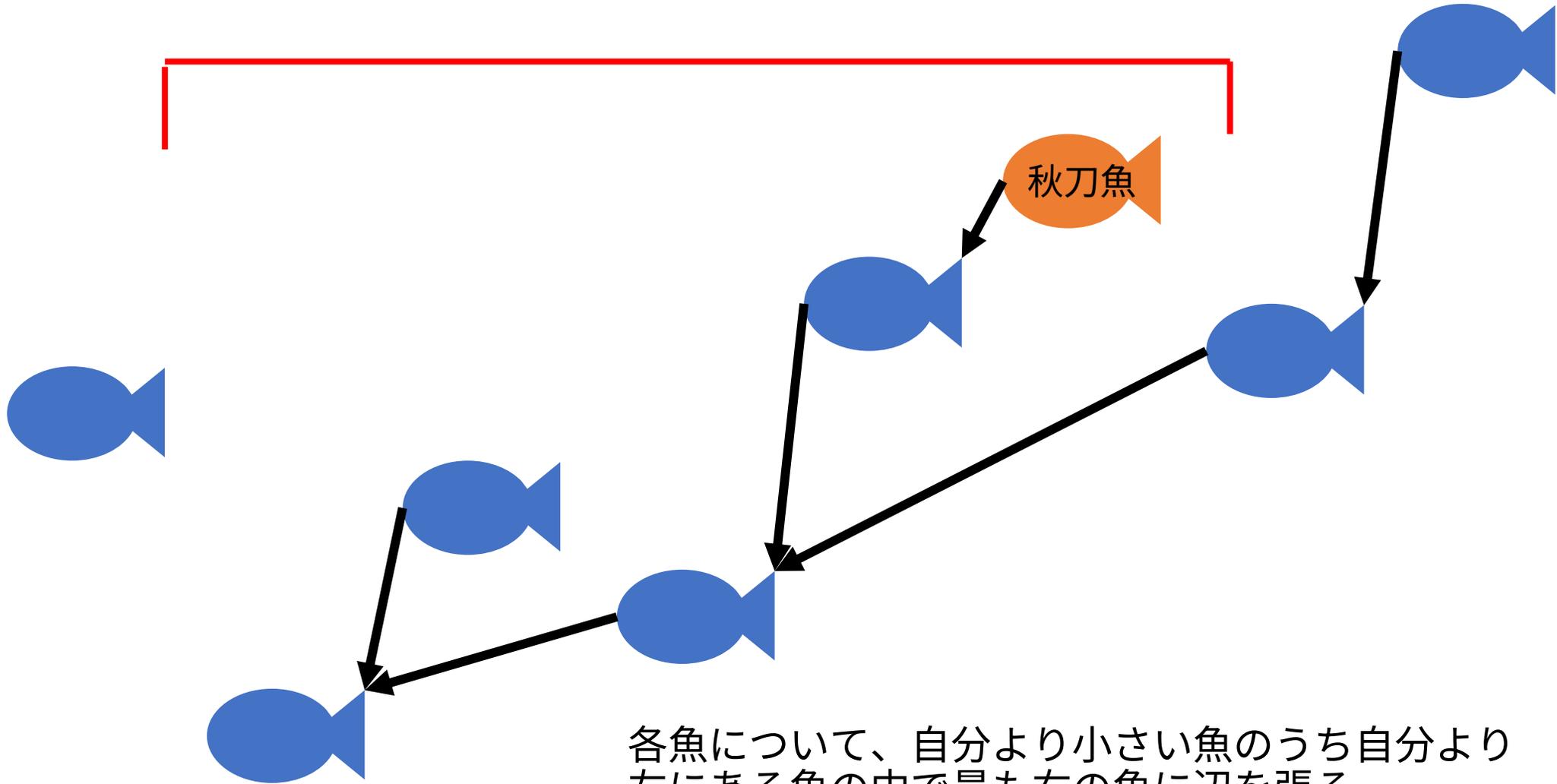


任意の区間クエリで秋刀魚だけを取り出
したい！



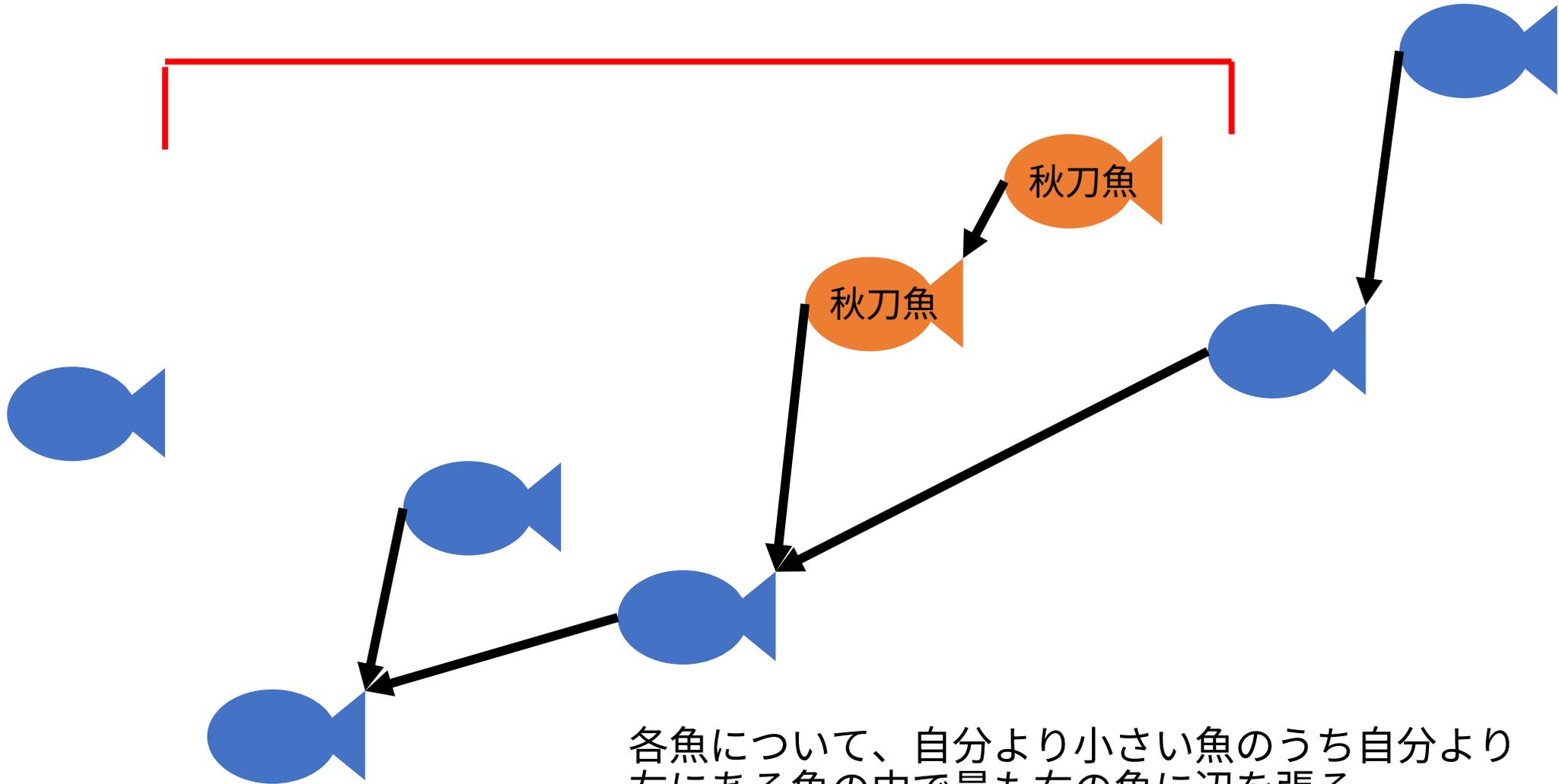
各魚について、自分より小さい魚のうち自分より左にある魚の中で最も右の魚に辺を張る

→有向森



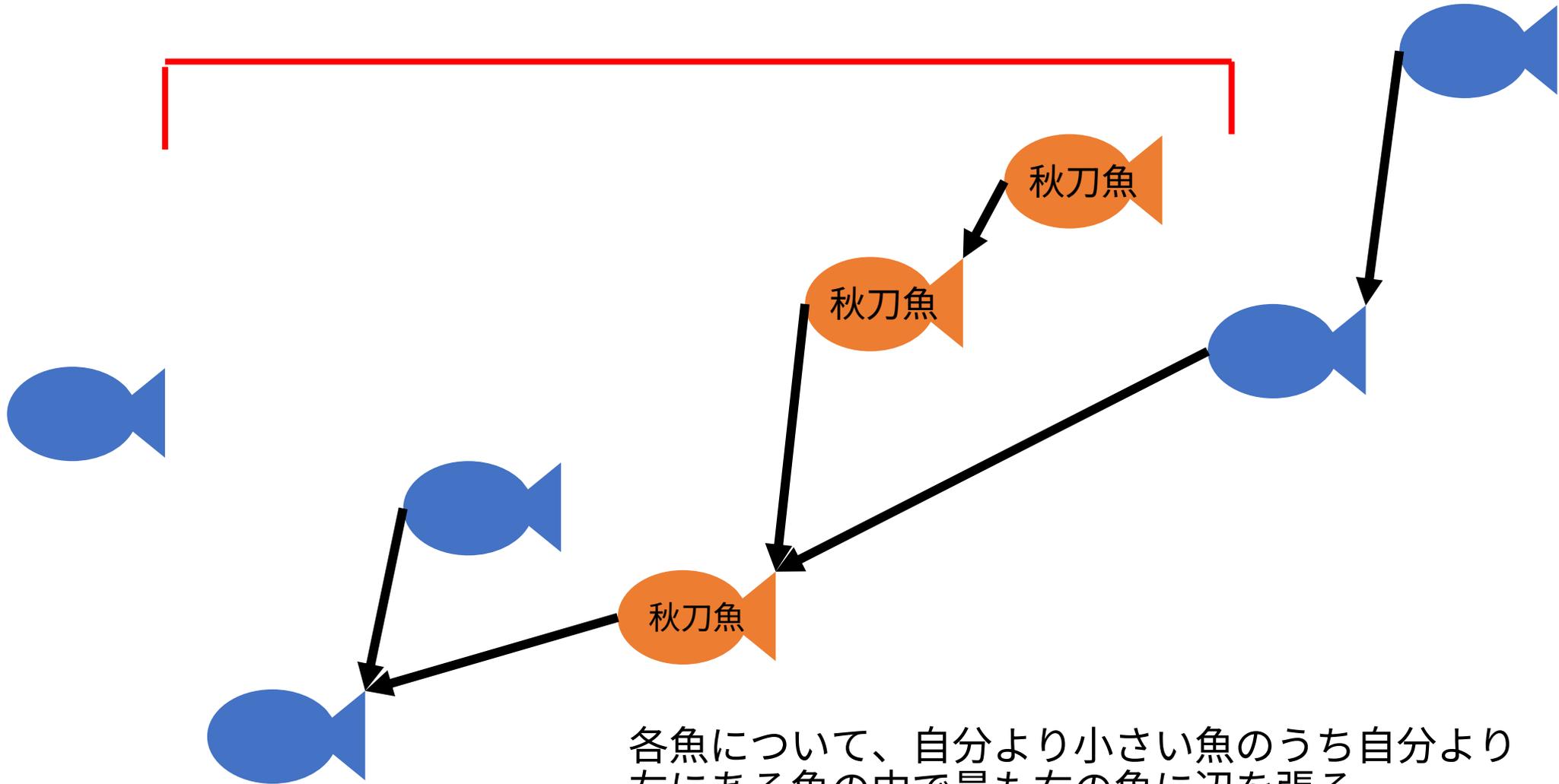
各魚について、自分より小さい魚のうち自分より左にある魚の中で最も右の魚に辺を張る

→有向森



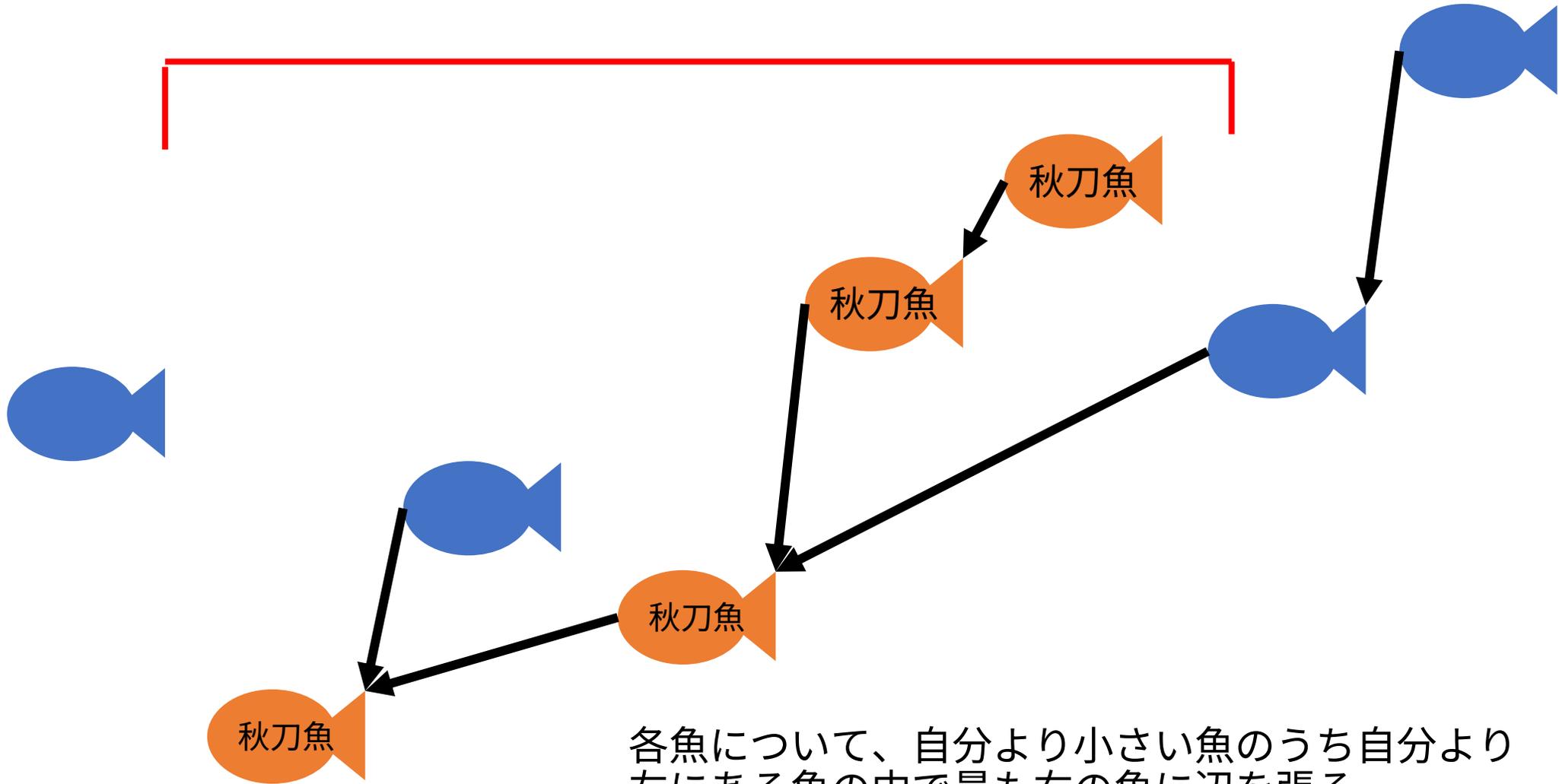
各魚について、自分より小さい魚のうち自分より左にある魚の中で最も右の魚に辺を張る

→有向森



各魚について、自分より小さい魚のうち自分より左にある魚の中で最も右の魚に辺を張る

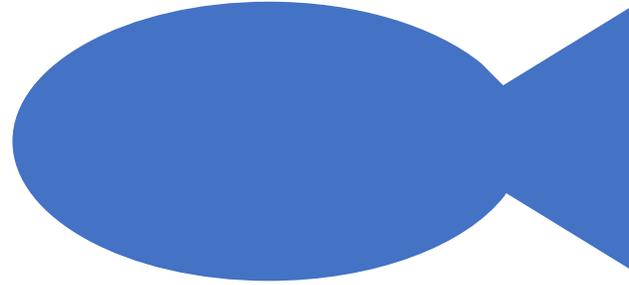
→有向森



各魚について、自分より小さい魚のうち自分より左にある魚の中で最も右の魚に辺を張る

→有向森

愚直に辺をたどっていくとクエリあたり $O(N)$ がかかってしまう



出次数が1以下のグラフで高速に辺をたどる時の典型テク

• • • ダブリング

$O((N+Q)\log N)$ で小課題3が通る

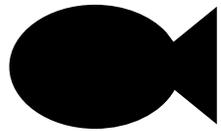
(別解)

データ構造を使わない選択肢

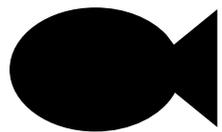
クエリは任意の順番で処理してよい

Rの昇順でクエリを処理するとどうなるか？

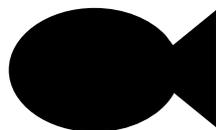


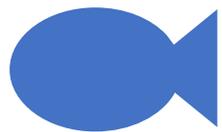


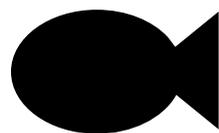
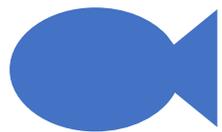


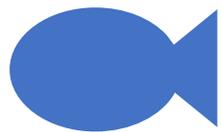


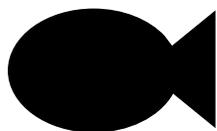
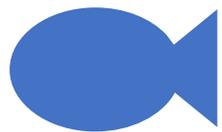


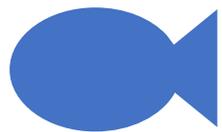


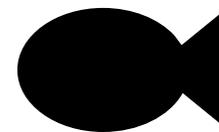
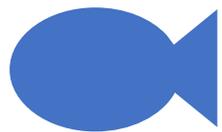


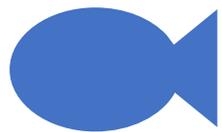


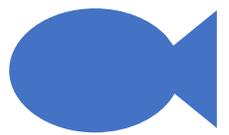
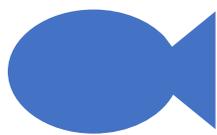
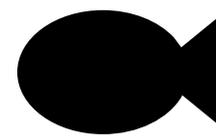














Rを右に移動させつつ、秋刀魚の一覧をスタックで管理しておく

クエリは適宜処理

秋刀魚の管理は合計で $O(N)$

スタックに累積和の情報も持たせておくと
クエリの処理は二分探索で各 $O(\log N)$

```

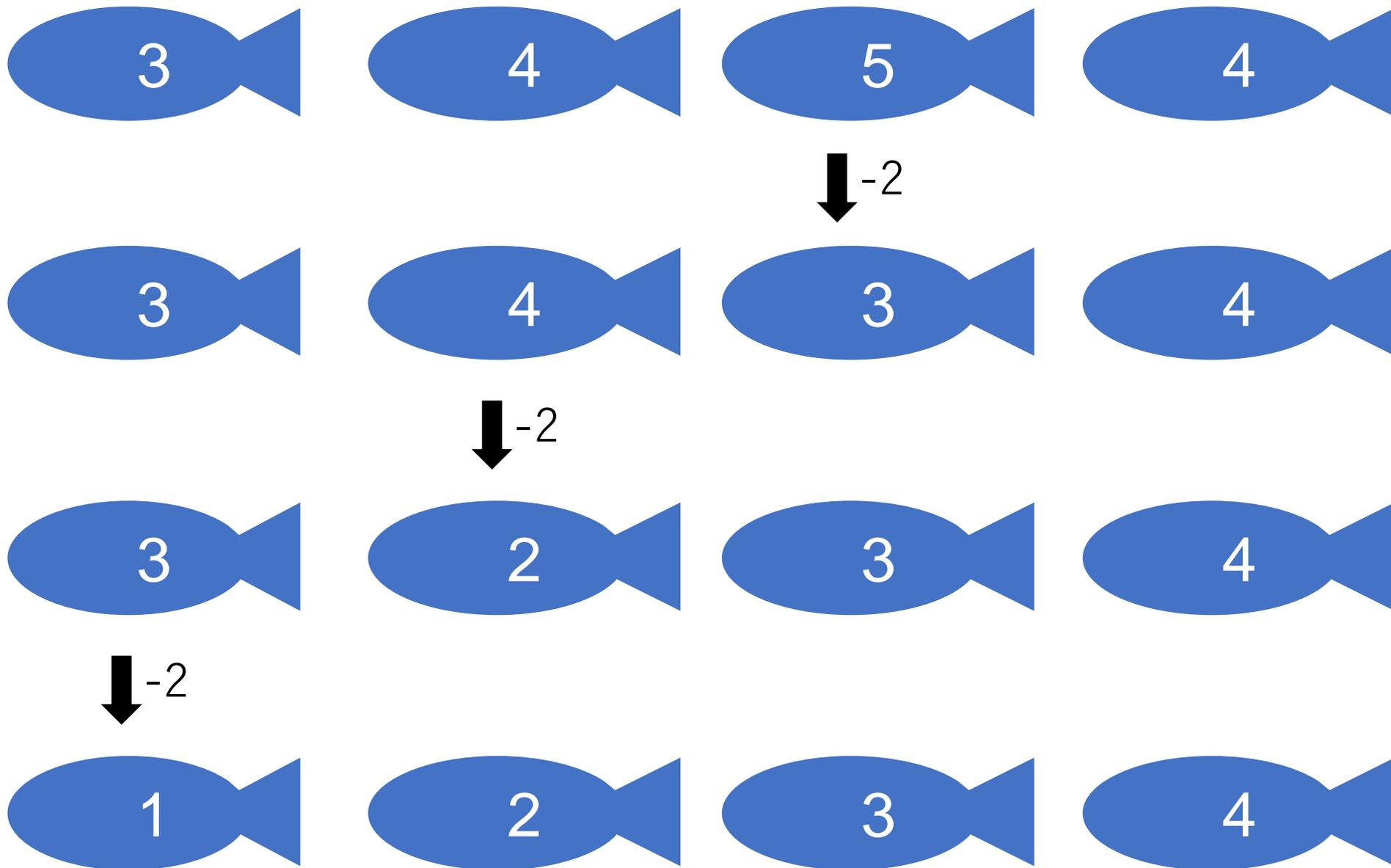
vi sol(ll n,ll q,vi v,vvp query){
    vi ans(q),rui(n+1);
    rep(i,n)rui[i+1]=rui[i]+v[i];
    vp st;st.pb(-1,0);
    rep(i,n){
        while(st.size()>1&&v[st.back().fi]>=v[i])st.pop_back();
        st.pb(i,st.back().se+v[i]*(i-st.back().fi));
        for(auto x:query[i]){
            ll idx=lb(st,P(x.fi,-inf));
            ans[x.se]=st.back().se-st[idx-1].se-v[st[idx].fi]*(x.fi-1-st[idx-1].fi);
            ans[x.se]=rui[i+1]-rui[x.fi]-ans[x.se];
        }
    }
    return ans;
}

```

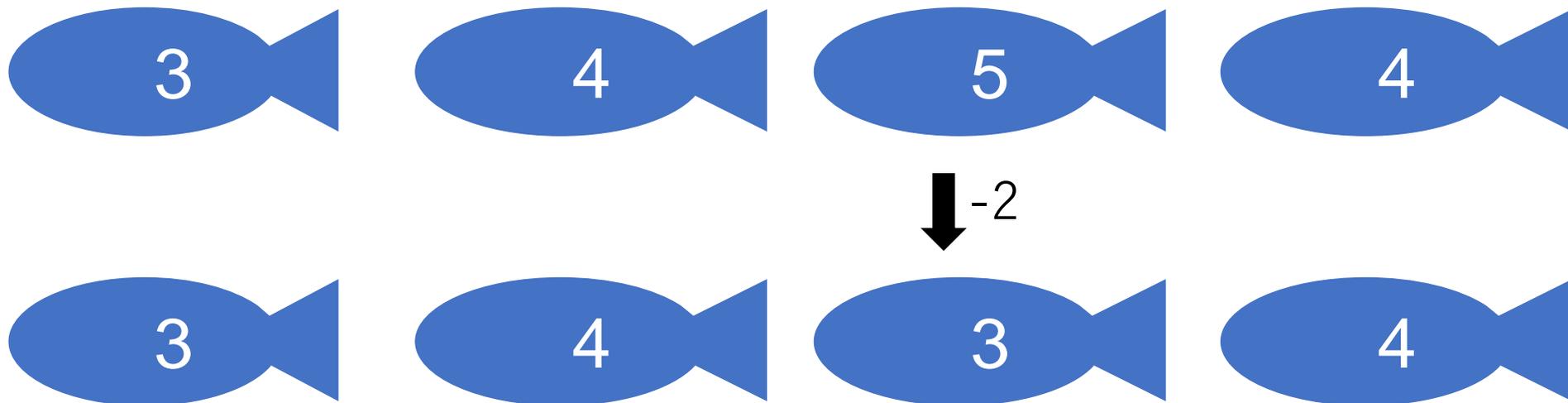
小課題 4

Cが単調減少

D=2



D=2

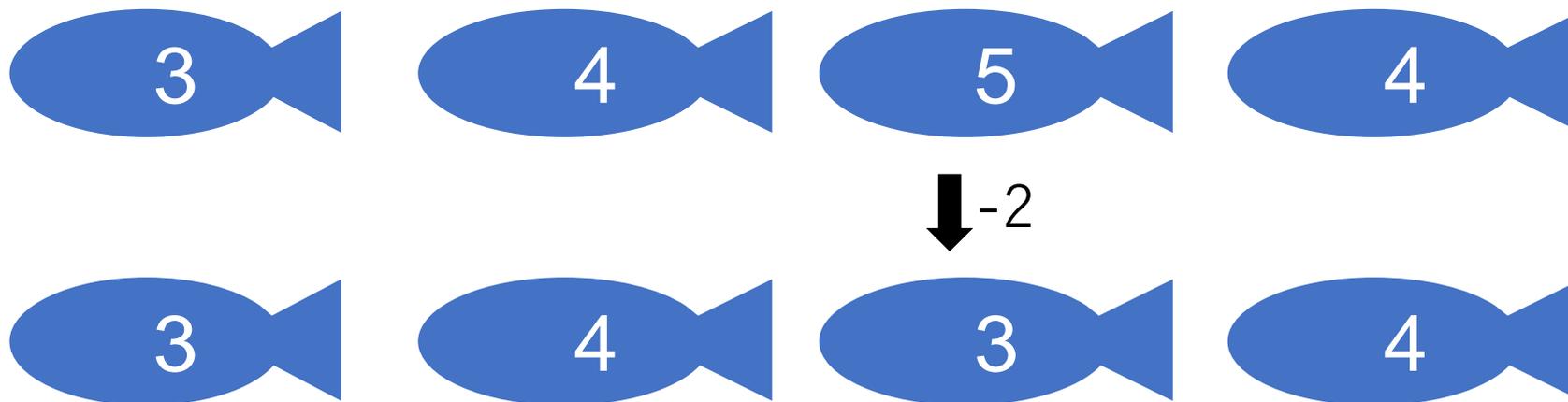


D=1の時にはなかった余計なIQの減少

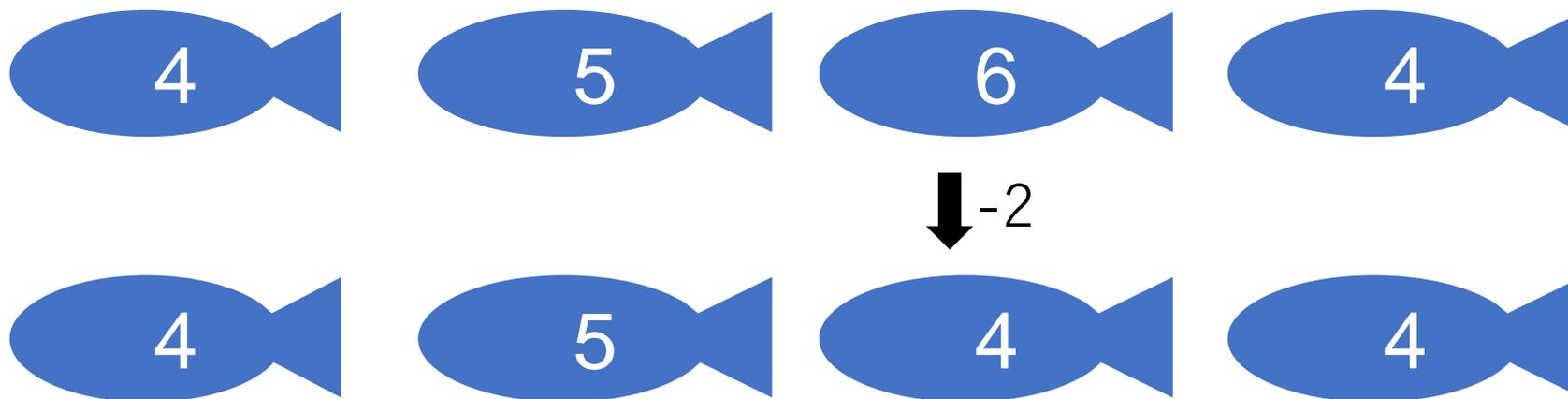
→なんとかしたい

とりあえず不可能なケースは後回し

D=2



元の数列を同値変形



全部の値がDの倍数になるように同値変形できる！

D=2



最後に全部 D で割れば $D=1$ の場合に帰着される

小課題4では元の数列が単調減少だから同値変形後も単調減少

操作後はすべての魚のIQが右端の魚のIQに一致するため、簡単に操作回数が求まる

左端の魚に対する操作回数が分かるため、同値変形前の数列を用いて不可能かどうか判定できる

```
for(ll i=n-2;i>=0;i--){  
    if(a[i]>=a[i+1])v[i]=v[i+1]+(a[i]-a[i+1]+d-1)/d;  
    else v[i]=v[i+1]-(a[i+1]-a[i])/d;  
}
```

小課題 5

追加の制約なし

小課題3と小課題4の解法を組み合わせます

AC !

(注意点)

魚のIQの最大値は 10^{12} なのでオーバーフローに気を付けましょう

得点分布

100点：9人

64点：3人

48点：1人

44点：1人

36点：1人

29点：2人

20点：1人

16点：5人

9点：4人

0点：0人

0点：0人