

# ABC 153 解説

writers: kyopro\_friends

2020 年 1 月 26 日

*For International Readers: English editorial will be published in a few days.*

## A: Serval vs Monster

求める答えは  $H/A$  の切り上げです。ceil 関数などを使ってもよいですし、 $(H + A - 1)/A$  の切り捨てにより求めることもできます。

C 言語での実装例は次のとおりです。

---

```
1 #include<stdio.h>
2 int main(){
3     int h,a;
4     scanf("%d%d",&h,&a);
5     printf("%d\n",(h+a-1)/a);
6 }
```

---

このほか、「 $H$  が 0 以下になるまで  $A$  を引く」というループ処理で解くこともできます。

## B: Common raccoon vs Monster

全ての必殺技を 1 度ずつ使ってモンスターの体力を 0 以下にできれば良いです。つまり、 $H - (A_1 + A_2 + \dots + A_N)$  が 0 以下のとき Yes、そうでないとき No となります。

C 言語での実装例は次のとおりです。

---

```
1 int main(){
2     int H,N;
3     scanf("%d%d",&H,&N);
4     for(int i=0;i<N;i++){
5         int A;
6         scanf("%d",&A);
7         H-=A;
8     }
9     if(H<=0)puts("Yes");
10    else puts("No");
11 }
```

---

## C: Fennec vs Monster

体力の高い方から  $K$  体のモンスターに必殺技を使い、残りのモンスターに攻撃するのが最適です。  
モンスターを体力の順にソートすることで  $O(N \log N)$  で解くことができます。

## D: Caracal vs Monster

体力が  $H$  のモンスター 1 体に勝つために必要な攻撃回数を  $f(H)$  とします。このとき

$$f(H) = \begin{cases} 2 \times f(\lfloor H/2 \rfloor) + 1 & (H > 1) \\ 1 & (H = 1) \end{cases}$$

となります。この定義に従って再帰的に計算することで答えを求めることができます。

## E: Crested Ibis vs Monster

次のような DP を考えることで解くことができます。

$DP[i]$  = モンスターの体力を  $i$  減らすため消費する魔力の最小値

同じ魔法を複数回使うことができるので、DP 配列の更新順序には気をつけてください。答えは  $\min_{H \leq i < H + \max A_i} DP[i]$  になります (モンスターの体力を  $H + \max A_i$  以上減らすには、 $H$  以上減らしてからさらに魔法を使う必要があるので無駄です)。この部分は、「 $i \geq H$  であるような部分について予めまとめた状態で DP 配列を持つ」という実装の工夫により簡略化することもできます。計算量は  $O(NH)$  です。

なお、答えは最大で  $H * \max A_i$  程度になるので、次のような DP では解けないことに注意してください。

$DP[i]$  = 魔力を  $i$  消費したときに減らせるモンスターの体力の最大値

## F: Silver Fox vs Monster

モンスターたちはあらかじめ座標の昇順にソートされているとしてよいです。座標が小さい方を”左”、大きい方を”右”という向きで呼ぶことにします。

最も左にいるモンスターを倒すためときには、できるだけ右で爆弾を使い、他のモンスターを多く巻き込むのが良いです。そのようにして左側から貪欲に倒していくことが最適になります。

尺取り法や二分探索などを用いて、各モンスター  $i$  について「 $X_j - X_i \leq 2D$  となる最大の  $j$ 」を計算しておくことで、モンスター  $i$  を左端として攻撃するときどこまで右側のモンスターを巻き込むことができるかわかります。あとは、累積和などを用いて既に減らした体力量を管理することで答えを求めることができます。

最初に座標順にソートする部分がボトルネックとなり、計算量は  $O(N \log N)$  です。