2 ビリヤード (Billiards)

Author : 井上 誠大

小課題 1

この小課題では，全ての i について $P_i = -1$ である．よって，ビ太郎は $A_i \leq X$ なるボール i のうち i が最大となるボールをひとつだけ落とせばよい．

小課題 2

この小課題では，ボールを必ずボール 1，ボール 2，...，ボール N の順で落とす必要がある．このとき，ボール i を落とすために必要な集中力の合計は $A_1 + A_2 + \cdots + A_i$ である．よって，$A_1 + A_2 + \cdots + A_i \leq X$ を満たす最大の i について，ボール 1，ボール 2，...，ボール i を落とせばよい．

小課題 3

全てのボール i について，ボール i を落とすために落とす必要のあるボールを遡っていき (ボール i を落とすためにはボール $j = P_i$ を落とす必要があり，ボール j を落とすためにはボール $k = P_j$ を落とす必要があり，... という要領)，それらのボールを落とすのに必要な集中力の合計が X 以下ならばボール i を落とすことができるという方法で判定を行えばよい．$P_i < i$ なので，遡る過程において N ステップ以内に $P_i = -1$ なるボールに到達して停止する．時間計算量は $O(N^2)$ である．

小課題 4

i が小さい方から順に，ボール i を落とすために必要な集中力 $\mathrm{dp}[i]$ を動的計画法により求める．この $\mathrm{dp}[i]$ は次の通りに求めることができる．

- $\mathrm{dp}[i] = A_i$ ($P_i = -1$ である場合)
- $\mathrm{dp}[i] = A_i + \mathrm{dp}[P_i]$ ($P_i \neq -1$ である場合)

あとは，$\mathrm{dp}[i] \leq X$ なる最大の i を求めればよい．時間計算量は $O(N)$ である．

小課題 5

小課題 3 同様に，ボール i を落とすために落とす必要のあるボールを遡っていく．しかし，遡っていく過程で，ボール k を落とすためにボール k を落とすことが要求されることがある (この k は i と同じであるとは限らないことに注意せよ)． このような循環依存が出現した場合，ボール i を落とすことはできない．
この判定は，今までに落とすことを要求されたボールの集合 S を保持し，ボール i からボール P_i へと遡っていく際に P_i が S に含まれるかどうかを判定すればよい．この判定はバケットソートの要領で $O(1)$ で実現可能である．
この判定を小課題 3 のプログラムに書き加えることで，時間計算量 $O(N^2)$ でこの小課題に正解することができる．

満点

小課題 5 同様に，ボール i を落とすために落とす必要のあるボールを遡っていきながら循環依存を検知し，各ボール i について，「ボール i を落とすことができるか」「落とすために必要な集中力 dp[i] 」の 2 つを動的計画法により求める． これは，以下のようなアルゴリズムで実現可能である．

- $P_i = -1$ であるなら，ボール i を落とすことができ， $\mathrm{dp}[i] = A_i$ である．
- そうでなく P_i へと遡る際に循環依存が検出されたなら，ボール i を落とすことはできない．
- そうでなく P_i が落とすことのできるボールであると判明しているなら，ボール i を落とすことができ， $\mathrm{dp}[i] = A_i + \mathrm{dp}[P_i]$ である．
- そうでなく P_i が落とすことのできないボールであると判明しているなら，ボール i を落とすことはできない．
- そうでないなら，ボール P_i について先に求解し，その結果を利用してボール i に関して「ボール i を落とすことができるか」と「 $\mathrm{dp}[i]$ 」とを求める．

このような手続きを踏むことで，ボール i からボール P_i に遡る行為はアルゴリズム全体で高々 1 度しか行われない．よって，時間計算量 $O(N)$ でこの問題に正解できる．