パラレルh

原案: itigo

解説: itigo

インテラクティブの実装:monkukui

３３10の解き方

試しに入力が3 3 10の時を考えましょう

３個の数字に1~3ずつ足して10を二本先取で勝ち

７７6を言うことができたら必勝←理由は先のスライドで

７７6は合計で20のため三つの数の合計を「20言ったら勝ちゲーム」同様に4の倍数に保持すれば勝ち

原則

サンプル2で見た通り、このゲームは基本的に「全て数の合計のmod(k+1)」がある値mとなる手を作ることが勝ち確手となり、それ以外の値しか作れない（要はmを押し付けられた）時負け確手である

このmが0の時後手必勝、それ以外が先手必勝となる。

サンプル2の場合は7 7 6が勝ち確となるためm=0であった

以下のスライドはこのmを求めることを目的と進める。

一般化①　前提法則

このゲームは自分のほうがhを言った回数(以下得点量)が少ない状態で相手にターンを渡した瞬間負け確定

↓why

相手はh-k以上を作らないよう維持しつつ、全てh-k-1となったときはh-kを二つ作れば絶対に得点が抜かれることはないから

例: 5 3 10の時に相手に1点取られたとすると自分は

10 6 6 6 6　や 10 7 7 6 6　を作ればよい

一般化②　h-k交換

このゲームには「h-k交換」という(いちごくんが勝手に命名)基本戦略が存在する

これは例えばサンプル2の必勝手順である7 7 6の「7 7」の部分を指していて、「h-kを偶数個作る」という操作をしたとき続く偶数ターンはh-kにk足してhを作る操作が最善(次のスライドで説明)となるため、実質的に7 7 6を作った瞬間10 10 6となったと考えることができ、h-k交換した部分をゲームから除外して相手のターンにすることができる。

つまり、h-k交換を行った結果相手に負け確の手を押し付けると自分は勝てるためサンプル2は7 7 6で勝ち確となる。

一般化③　h-k交換、考察

h-k交換をされたとき本当に「hを作る」以外の操作はあり得ないか

Case1.h-kをh-k+aにする

例:7 7 6 →8 7 6

この場合、素直に10 7 6などといったhを作る操作を行うと

10 10 6と返されてしまい、負け確の6を押し付けられる。

しかし、8を10にするならあと1追加可能であり、これを6に足して10 7 7とする、つまりh-k交換の先を勝ち確手に変えることで対応できる。

一般化④　h-k交換、考察2

Case2:h-k交換以外のところをいじる

例:7 7 2 → 7 7 3

h-k交換以外のところは勝ち確手としてあるため、この行為は勝ち確手を負け確手に変更しているといえる。

しかし、これは負け確手を勝ち確手に戻す(7 7 6など)や、h-k交換を自分から始められるので、終わりも自分のターンとなり、結局自分が勝ち確手を言える(10 7 3 → 10 10 3 → 10 10 6など)

一般化⑤　mを求める

以上より偶数個のh-kをhにするという操作は実質的に「無」と言っていいので、スライド3で言った「全ての数の合計」は、hもh-kも同様にh-kだとして扱った時の合計と考えることができる。こうすればh-k交換の回数に関係なくmを不変な値と見れる。

例:10 10 6なら合計は20

一般化⑥　mを求める2

さて、mを求めたい

直感的に明らかにh h h h h h h-k-1のような形を言えば必勝であるため

これの「全て数の合計」は(n-1)*(h-k)+h-k-1であるため

m=((n-1)*(h-k)+h-k-1)%(k+1)であると思われる。

kが奇数の時はこれで正しい

しかしkが偶数の時、このようにはならない

kが偶数の時

例えば入力が5 2 5の時

3 3 3 3 2を言えば必勝の思われる。

実際3 3 3 3 2が言えれば必勝なのだが、3 3 3 3 2を言うためにmを2で保持しようとすると途中3 2 2 2 2を言わなくてはならない。

しかし、これを言うと5 2 2 2 2と返されてしまい、前提法則よりもう勝てなくなる。

要はこのゲームにはh-kを作るときは必ず偶数個にするという制約があるのだ

kが奇数ならk+1が偶数のためこの心配はいらない

kが偶数の時2

ではkが偶数の時のmを求めたい

試しに入力が5 2 5の時の2 2 2 2 2を考えると、ここからh-k交換のみを行って5 5 5 5 2を言えれば勝ちなのである。

h-k交換は偶数しか使えないことより、この問題は「1~kのうちの偶数だけを使ってn-1言ったら勝ちゲーム」となる。

例なら2のみ使って4言ったら勝ちゲームのため自明に後攻が勝ちで2 2 2 2 2を言った人が必勝となる。

よってこの場合のmは10%3=1より1である

一般にはm=(n*(h-k-1)+((n-1)/2)%(k/2+1)*2)%(k+1)になる

kが偶数のとき、m不定じゃね？

タイトルの通り、例えば

２２２２２はm=1、５５５５２はm=2である

これはkが偶数の時、目指す必勝手はh-k未満の数字の数によって変化するためである

具体的にはh-k交換をk+2個分行う度にｍは(合計がk+2増えるので)1増える。

これは一見危険に見えて、もし必勝手順を踏んでたとしても相手にうまくh-k交換をされながらｍをm+1に合わせられると負ける可能性があるかに思われる。

でも安心

しかし、そんなことは起きない。

なぜなら「h-k交換をしてmを1増やす」という操作をするには最低合計を2増やす必要があるが、自分が合計をmでキープしてるとき、相手が「mを1増やしたうえでm+1を獲得する」には合計を1増やすしかなくこれではh-k交換が行えないからだ。

mを2以上増やそうとしても、最低2+(k+2)合計を増やすことが要求されるため不可能である。

結局、場にh-k未満が残っている状態で自分がh-k交換を仕掛けることをしない限りこのmの変化によって必勝を譲ることは起きない。

h<=kのとき

先手必勝

先手で取れる限りの得点を取り、それ以外の値に触れないを繰り返しているだけで勝てる。

まとめ

k<hの時

k%2=0の時m=(N*(h-k-1)+((N-1)/2)%(k/2+1)*2)%(k+1)

k%2=1の時m=((n-1)*(h-k)+h-k-1)%(k+1)

を維持する(ただしNはh-k未満の値の数)

k>=hの時

先手必勝で、貪欲に取れるだけ取る

想定WA

- Ｋ＞Hに対応できない
- Kが偶数に対応できない
- 3 3 10のケースで7 8 6を言われたときに7 10 6を返す(7 10 7を返さなくてはいけない)
- Kが偶数のときにh-k交換を仕掛ける(mがずれてパラレルhに取られます)

想定コード

おそらくパラレルHのコードが欲しいと思われるので公開します

https://onlinejudge.u-aizu.ac.jp/beta/review.html#ACPC2020Day3/4860622