Анализ

Това е трудна задача, макар реализацията на решението да е сравнително проста.

Всички решения разглеждат тестовете в един входен файл поотделно, и съответно обозначените сложности са за един тест.

При L > B е очевидно, че Лора печели играта. Решенията в този анализ приемат, че L ≤ B.

Подзадача 1

В тази подзадача е достатъчно всякакво пълно изчерпване. Единствената трудност е, че технически играта може да продължи вечно. Това може да се игнорира като се приеме, че игра с твърде много ходове е равенство (в следващите подзадачи ще стане ясно, че това е наистина така)

Подзадачата позволява и hardcode-ване на отговорите.

Подзадача 2 – $O(N^4)$

Стандартно за такъв тип задача е да се опитаме да дефинираме печеливши, губещи и равни позиции. Логиката не е по-различна от дефиниране на печеливши и губещи позиции. Формално:

- Позиция е печеливша за един играч ако той има ход, който пренася играта в позиция губеща за другия играч
- Позиция е губеща за един играч ако всеки ход, който може да направи, пренася играта в позиция печеливша за другия играч
- Позиция е равенство ако не е нито печеливша нито губеща

Тези дефиниции работят добре за нециклична игра, затова трябва да представим дадената ни игра като крайна. Представените решения се справят с този проблем по различни начини.

Нека дефинираме позиция в играта. В случая Лора и Боби не играят по идентични правила и затова трябва да пазим информация и за двамата. Можем да опишем една позиция чрез следната информация:

- Позиция на Лора
- Позиция на Боби
- Играч на ход

Така стейтът е с големина $O(N^2 * 2) = O(N^2)$. Проблем е обаче цикличността – възможно е след няколко хода от дадена позиция да се върнем отново в нея. Едно решение на проблема е да добавим допълнителен стейт “номер на текущия ход”. Така очевидно играта става ациклична, но не е ясно колко хода може да продължи една игра. Интуитивно, една игра не може да продължи твърде много ходове. Оказва се, че е достатъчно да считаме всяка игра продължила над 3N хода за равенство. Това не е строга граница, но е важно да отбележим, че ако считаме всяка игра продължила над N хода за равенство това не е правилен подход. При игра, в която имаме L=1, B=2, K=1, победител е Боби, но Лора може да “забави” играта до общо 2N хода.

Така описваме една позиция с $O(N^3)$ информация. За да изчислим една позиция трябва да разгледаме всички възможни ходове от нея. Това е O(L) или O(B) в зависимост от това кой е на ход, но и в двата случая е най-много O(N). Получаваме сложност O(N^4).

Подзадача 3 – O(N^3)

За тази подзадача човек трябва да се замисли, че макар играта да е циклична, тя е ***почти*** ациклична.

Нека позицията на Лора в даден момент е P_L, а позицията на Боби P_B. Да разгледаме как се изменя стойността $MAX(P_L, P_B)$ при даден ход. Ако никой от играчите не бутне другия назад, то просто едната стойност от двете се увеличава, и съответно $MAX(P_L, P_B)$ или се увеличава или не се променя. От друга страна ако единият играч бутне другия, то тогава бутнатият е бил по-напред и е задавал стойността на $MAX(P_L, P_B)$, но сега бутналият го е на тази позиция и стойността отново не се е променила. Така можем да заключим, че тази стойност е ***ненамаляваща*** в процеса на играта. Цикличността идва от това, че може да не се променя.

Нека разгледаме три поредни хода. Ако стойността $MAX(P_L, P_B)$ не се изменя, то това значи че и трита хода са били бутане. От друга страна можем да забележим, че ако три поредни хода са бутане, то имаме повторена позиция предизвикана от двамата играчи. Ако някой от играчите може да спечели, то той няма причина да участва в 3 поредни бутания. Следователно можем вместо бройка направени ходове да държим просто броя поредни бутания в последните ходове. Позиции с 3 поредни бутания могат да бъдат обявени за равенство. Забележете, че 2 поредни бутания **не** означават равенство.

Това редуцира описването на позиция до O(2*3*N^2) = O(N^2).

Изчислението не е подобрено, съответно получаваме общо O(N^3)

Подзадача 4 – O(N^2)

За тази подзадача се изискваше трудната стъпка в задачата.

Нека си представим, че играчите можеха да стъпват на едно и също място без да се бутат. В такъв случай задачата е елементарна – всеки просто ще прави най-големия възможен за него ход, докато някой не стигне финала. Интуитивно, бутането не би трябвало да усложни задачата толкова много.

Нека направим няколко наблюдения за това как протича една игра и за това какви ходове правят играчите.

- След всеки свой ход Лора иска да се намира пред Боби. Тъй като $L \leq B$, то ако в даден момент Лора се намира зад Боби и е негов ход, то той може да спечели просто правейки само максимални ходове.
- Ако след максимален свой ход Боби може да се отдалечи на разстояние над L пред Лора, то той ще направи това и ще спечели, тъй като след следващият ход на Лора тя ще е зад него. Такъв ход на Боби ще наричаме "***избягване***" и съответно ако Боби може да направи такъв ход казваме, че може да "***избяга***"
- Ако след своя ход Боби иска да се озове зад Лора, то той има смисъл да отиде само точно едно поле зад нея. Всяка друга позиция зад Лора просто би му дала по-малко възможности . Ход, в който Боби отива точно едно поле зад Лора ще наричаме "***пасивен ход***"

Очевидно интересните позиции в играта са тези директно след бутане на някой от играчите. Интересна позиция се оказва и такава непосредствено след пасивен ход. В такъв случай нека дефинираме няколко функции, които просто отговарят на по-интересни стейтове:

$pushL_i =$ резултатът от позицията, в която Лора е на ход на поле $MAX(1, i-K)$, а Боби е на поле i

$pushB_i =$ резултатът от позицията, в която Боби е на ход на поле $MAX(1, i-K)$, а Лора е на поле i

$passive_i =$ резултатът от позицията, в която Лора е на ход на поле i, а Боби е на поле $i-1$

Очевидно тези позиции отговарят съответно на позициите веднага след бутане на единия, както и позицията веднага след пасивен ход.

Да си представим обаче, че тези функции са сметнати предварително и искаме да проверим дали Лора може да спечели от дадена позиция. Тогава можем да симулираме игра "без бутане" подобно на грийди подходът:

- Ако Боби може да избяга, то той прави това. В противен случай той отива на най-далечната позиция i, за която $pushB_i$ не е победа за Лора.
- На свой ход Лора отива до най-далечната позиция i, за която $pushL_i$ е победа за Лора, както и $passive_i$ е победа за Лора.

По този начин не е нужно изобщо да разглеждаме бутания или пасивни ходове, тъй като никой от играчите няма да се постави в позиция, в която бутане би развалило целта му. Симулацията за да проверим дали Боби може да спечели дадена позиция е аналогична.

Да забележим всъщност, че пресмятането на тези функции решава и задачата, тъй като отговорът на задачата е $pushL_1$. Остава само някак да пресметнем стойностите на тези функции.

Нека сме пресметнали $pushL_j$, $pushB_j$ и $passive_i$ за всички $j > i$ и сега искаме да пресметнем $pushL_i$, $pushB_i$ и $passive_i$.

Генералният подход за пресмятане на дадена позиция е да тестваме две хипотези. Едната хипотеза е, че Боби може да спечели от тази позиция, а другата, че Лора може да спечели от тази позиция. Ако имаме отговори на двете хипотези, то лесно имаме и отговор за цялата позиция – ако никой не може да спечели, то позицията е равенство, а в противен случай печели този който може. Предимството, което получаваме от това, че фиксираме хипотеза е, че за всяка позиция знаем със сигурност кой от двамата играчи би искал да се стигне до нея. Например ако тестваме хипотезата, че Лора може да спечели, то всички позиции различни от победа за Лора стават "победа" за Боби в контекста на хипотезата.

Пресмятането на всяка от стойностите $pushL_i$, $pushB_i$, $passive_i$ става като просто тестваме двете хипотези започвайки от съответната позиция и използвайки симулацията без бутания посочена по-горе.

Малък проблем е, че на практика можем да имаме преходи между тези 3 позиции, а очевидно не можем да знаем стойностите им преди да ги пресметнем. Например от позицията на $pushB_i$ при $K \neq 1$ можем да преминем в тази на $passive_i$. Подобно можем да минаваме от $pushB_i$ към $pushL_i$ и обратно. Тази малка цикличност можем да оправим ръчно по следният начин:

- В симулацията приемаме, че преход от една от трите позиции съответстващи на $pushL_i$, $pushB_i$, $passive_i$ не можем да преминем към друга от тях. Тоест при симулация на $pushB_i$ първият ход не може да е бутане или пасивен ход и при симулация на $pushL_i$ първият ход не може да е бутане.

- След пресмятане на стойностите е нужно да направим корекции поради възможността за преходи между тях. Корекциите са две:
    - Тъй като от позицията за $pushB_i$ Боби може да преминем към $passive_i$, то ако $passive_i$ е по-добра за Боби взимаме същият отговор за $pushB_i$. Така например ако симулацията даде $pushB_i = draw$, но $passive_i = win\_Bobi$, то слагаме $pushB_i = win\_Bobi$.
    - Ако резултатът от $pushL_i$ е по-добър за Боби от $pushB_i$, то следва със сигурност и че $pushB_i$ е по-добро за Лора отколкото $pushL_i$. В такъв случай и двамата играчи започвайки от която и да е от двете позиции ще искат да преминат в другата. Това води до безкрайно прескачане и в такъв случай и двете позиции стават равенство. Във всеки друг случай и двете позиции запазват стойността си от симулацията.

Симулацията може да се имплементира за O(N) време. Пресмятайки дадените функции отзад напред получаваме обща сложност O(N^2).

Подзадача 5 – O(N)

Стъпката към линейно решение не е толкова голяма. Идеята е просто да използваме това, че ако нямаме бутане, то това един играч да е по-напред винаги е предимство. В такъв случай можем да дефинираме функциите:

$F_i =$ най-задната позиция, на която може да се намира Боби, така че при симулация на хипотезата, че Лора може да спечели, започвайки с Лора на ход и на позиция i, Боби стига финалът пръв.

$G_i =$ най-задната позиция, на която може да се намира Боби, така че при симулация на хипотезата, че Боби може да спечели, започвайки с Лора на ход и на позиция i, Боби стига финалът пръв.

Ако пресметнем тези функции то лесно можем да предсказваме резултата от дадена хипотеза бързо. Например ако Лора е на позиция P_L и на ход, а Боби е на позиция P_B и искаме да тестваме хипотезата, че Лора може да спечели, то е нужно просто да проверим дали $F_{P_L} > P_B$. Ако пък отново са на тези позиции, но Боби е на ход, можем да симулираме първият му ход и попадаме в позиция, в която Лора е на ход и която решаваме по същия начин.

В процеса на пресмятане отзад напред можем да пресмятаме и първите ходове от всяка симулация. Нека искаме да пресметнем F_i. Да предположим, че $F_i = x$. Тогава искаме да видим как би свършила симулацията с Лора на ход на позиция i, и Боби на позиция x. Това можем да направим просто като направим по 1 ход от симулацията за двамата, след което се озоваваме в позиция на Лора, за която F е сметнато и можем да проверим отговорът константно. Ако се окаже, че Боби стига финала първи, то знаем че $F_i \leq x$, а в противен случай $F_i > x$. Това позволява двоично търсене по x за логаритмично време, но можем да пресмятаме и за амортизирано константно време използвайки линейно търсене и фактът, че $F_i \leq F_{i+1}$. Функцията G се смята по аналогичен начин, просто за другата хипотеза. За да не се налага при пресмятане да използваме стойности, които още не са сметнати, то можем да приемем, че $F_i \geq i - 1$, тъй като Боби никога няма да е по-назад от 1 поле спрямо Лора при неин ход.

Така решението се реализира като едно обхождане от N към 1, поддържайки *няколко* стойности за всяка клетка.

Автор: Енчо Мишинев