Tư duy thuật toán

Lê Trí Đức

January 2024

Trong bài viết này, ta sẽ tập trung vào giải quyết một số bài toán tổ hợp theo hướng tư duy thuật toán.

Đầu tiên, ta cần phải hiểu thế nào là tư duy thuật toán. Thuật toán là một quy trình mô tả những chỉ dẫn chính xác và rõ ràng để thực hiện một công việc cụ thể nào đó. Mỗi quy trình sẽ gồm các bước khác nhau, chứa nhiều thao tác và mỗi thao tác như vậy góp phần đưa ta đến được lời giải của bài toán. Ví dụ trong đời sống ta cũng thường hay rơi vào những tình huống buộc ta phải sử dụng thuật toán để giải quyết, chẳng hạn như sắp xếp và phân loại tủ sách theo các thể loại khác nhau v.v...Hướng tư duy như vậy khá thú vị và cũng đồng thời giúp ta đơn giản hóa được vấn đề cần giải quyết, vì thuật toán có xu hướng thu thập thông tin một cách tuyến tính nên ta có thể từ từ giải quyết bài toán bằng cách phân loại và thực hiện các quy trình. Vậy thì sự khác biệt giữa phương pháp giải toán thông thường so với phương pháp thuật toán mà ta đã đề ra ở trên. Ở phương pháp thông thường, ta thường đi từng bước một từ giả thiết đến kết luận để xử lí bài toán. Trong khi đó, đối với phương pháp thuật toán ta chỉ cần chỉ ra được mỗi chuỗi các hành động cụ thể có xu hướng lặp lại hoặc rẽ nhánh và cho các đối tượng "chạy" là được. Tuy vậy phương pháp này vẫn còn nhiều hạn chế, đặc biệt là những bài toán mà ta không thể mô hình hóa các đối tượng để xử lí được mà cần các thao tác trừu tượng hơn.

Dưới đây là một số bài toán tổ hợp hay mà các bạn có thể thử giải.

Bài 1. Peter có 3 tài khoản ngân hàng, mỗi tài khoản chứa một số nguyên tiền đô la. Bạn ấy chỉ có thể chuyển tiền từ tài khoản này sang tài khoản khác sao cho số tiền được chuyển đúng bằng số tiền hiện có trong tài khoản nhận tiền. Chứng minh rằng Peter luôn có cách chuyển hết tiền vào hai tài khoản. Liệu bạn ấy luôn có cách chuyển tiền vào một tài khoản hay không?

Lời giải.

Ví dụ một cách chuyển là hợp lệ giữa 2 tài khoản như sau : tài khoản 1 có 50 đô còn tài khoản 2 có 10 đô. Bây giờ ta sẽ chuyển tiền từ tài khoản 1 sang tài khoản 2 và số tiền phải chuyển là 10 đô.

Đầu tiên, ta nhận thấy rằng để chuyển toàn bộ số tiền từ 3 tài khoản vào 1 tài khoản duy nhất thì trước đó sẽ còn lại đúng 2 tài khoản có tiền trong đó và bước tiếp theo ta sẽ chuyển toàn bộ số tiền từ tài khoản này sang tài khoản kia. Cuối cùng số tiền trong tài khoản cuối cùng được chuyển sẽ gấp đôi lên và sẽ là một số chẵn. Con số này cũng chính là tổng số tiền của cả 3 tài khoản cho nên nếu như ban đầu tổng số tiền ở 3 tài khoản là một số lẻ thì Peter không có cách nào để chuyển hết về một tài khoản.

Bây giờ ta sẽ chỉ ra một thuật toán để Peter có thể hết toàn bộ số tiền về hai tài khoản.

Gọi 3 tài khoản đó lần lượt là A, B, C và gọi số tiền trong đó theo thứ tự là a, b, c. Không mất tính tổng quát giả sử $a = \min\{a, b, c\}$. Ta sẽ tiến hành chuyển tiền từ hai tài khoản b, c sang tài khoản a. Xét $a \leqslant b \leqslant c$. Bây giờ ta sẽ chuyển dần từ B sang A. Mỗi lần chuyển số tiền luôn có dạng $2^i a$ cho nên ta nghĩ đến việc biểu diễn b và a thông qua phép chia Euclid và hệ nhị phân. Cụ thể ta xét $b = qa + r, 0 \leqslant r < a$ trong đó q có biểu diễn nhị phân là

$$q = m_0 + 2m_1 + ... + 2^k m_k$$

Nhưng vấn đề ở đây là, không phải chữ số nào của q trong hệ nhị phân cũng là 1 cho nên ta không thể chuyển liên tiếp được. Thế thì, mỗi chữ số bị khuất như thế ta sẽ dùng tài khoản C để bù lại. Cụ thể ta xét một thuật toán chuyển như sau: Ta xét lần lượt từng chữ số của q trong hệ nhị phân. Nếu $m_0 = 1$ thì ta chuyển từ B sang A. Còn ngược lại nếu như $m_0 = 0$ thì ta sẽ chuyển từ C sang A. Và trong quá trình này, ta luôn có thể chuyển được từ C sang A vì $b \leqslant c$. Lúc này ta sẽ tiếp tục chuyển cho tới khi chuyển được số tiền là qa từ b sang a. Tài khoản B còn lại số tiền là r và $r < a$. Do đó ta tiếp tục gọi $s = \min\{r, a', c'\}$ trong đó r, a', c' là số tiền của 3 tài khoản B, A, C lúc này. Ta tiếp tục tiến hành quá trình chuyển tiền như trên đồng thời chú ý rằng $s \leqslant r < a$ cho nên số tiền nhỏ nhất của 3 tài khoản sẽ giảm dần sau mỗi quá trình chuyển. Vậy suy ra phải có một thời điểm nào đó mà số tiền đó đúng bằng 0 và ta đạt được mục tiêu. Bài toán hoàn tất.

Nhận xét: Khi giải quyết các bài toán như trên, ta thường có xu hướng tìm kiếm các đơn biến bất biến, mà thông thường là đơn biến. Đơn biến này giúp ta chỉ ra rằng thuật toán sẽ dừng lại một lúc này đó và không thể tiếp tục vô hạn lần.

Bài 2. Có 100 anh em nhà gấu vào rừng hái dâu. Cậu em trẻ nhất hái được 1 quả. Cậu em trẻ thứ hai hái được 2 quả. Cậu em trẻ thứ ba hái được 4 quả. Và cứ như vậy, người anh hai cả hái được 2^{99} quả. Các anh em nhà gấu gặp một con cáo và được gợi ý cách chia dâu công bằng như sau: Cáo được chọn hai anh em gấu bất kì, lấy tổng số dâu của hai anh em. Nếu là số lẻ, cáo sẽ ăn một quả dâu rồi chia đều số dâu còn lại cho hai anh em. Nếu là số chẵn, cáo sẽ chia đều số dâu cho hai anh em. Cáo sẽ thực hiện chia cho tới khi tất cả các anh em gấu có số dâu bằng nhau. Hỏi số dâu ít nhất mà cáo có thể để lại cho 100 anh em nhà gấu là bao nhiêu?

Lời giải.

Đầu tiên, ta cần nhận xét rằng con cáo không thể ăn hết toàn bộ số dâu của các anh em nhà gấu. Nếu như sau một bước chia mà cả 2 con gấu đều nhận được số dâu là 0 quả thì lúc trước khi chia, tổng số dâu của cả 2 con gấu sẽ là 1. Điều này có nghĩa là sẽ có 1 con gấu không có quả dâu nào. Nhưng trong quá trình chia dâu ở trên không bao giờ xuất hiện một con gấu không có quả dâu nào được vì lúc đầu con gấu nào cũng có dâu. Ta chỉ cần để ý đến bất đẳng thức $\left\lfloor \dfrac{a_i + a_j}{2} \right\rfloor \geqslant \min\{a_i, a_j\}$ và trong số các con gấu thì số dâu luôn dương.

Do đó con cáo không thể ăn hết toàn bộ số dâu của các con gấu được.

Bây giờ ta vào vai con cáo và việc cần làm là tìm ra một thuật toán để ăn được nhiều dâu nhất có thể.

Đầu tiên, ta sẽ lấy số dâu của 2 con gấu đầu tiên. Tổng số dâu của 2 con gấu là 1+2=3 và là số lẻ nên ta sẽ trừ 1 đi. Lúc này sau bước chia đầu tiên 2 con gấu sẽ có số dâu bằng nhau và bằng 1. Khi đã bằng 1 rồi thì không có cách nào để giảm xuống 0 quả được theo nhận xét ở trên. Cho nên tạm thời ta không cần quan tâm tới 2 chú gấu này. Sau đó ta sẽ xét con gấu thứ 3 có 4 quả dâu. Lúc này ta xét con gấu thứ 2 và con gấu thứ 3 và tiến hành chia như trên thì 2 con gấu này sẽ có mỗi con 2 quả dâu. Để thuận tiện ta có thể ký hiệu $(a_1, a_2, ..., a_{100})$ là số dâu mà mỗi con gấu đang có theo thứ tự như vậy.

Sau bước chia dâu ở trên thì số dâu hiện tại là như sau $(a_1, a_2, a_3) = (1, 2, 2)$, tiếp theo ta lại chọn ra 2 con gấu a_1 và a_2 để tiến hành chia dâu thì sẽ được $(a_1, a_2, a_3) = (1, 1, 2)$ và chọn 2 con gấu a_2 và a_3 để chia. Cuối cùng sẽ được $(a_1, a_2, a_3) = (1, 1, 1)$.

Đến đây ta nảy ra ý tưởng là gì? Mỗi lần như vậy, ta sẽ ghép cặp từ trên xuống dưới từ từ để giảm số dâu mà mỗi con gấu đang giữ. Ta thử với con gấu a_4 có 8 quả dâu xem sao.

+Chọn con gấu a_4, a_3 và chia thì lúc này $(a_3, a_4) = (1, 8) \to (4, 4)$ sau đó giảm xuống tiếp $(a_2, a_3) = (1, 4) \to (2, 2)$ và tiếp tục $(a_1, a_2) = (1, 2) \to (1, 1)$ và quay trở lại $(a_2, a_3) = (1, 2) \to (1, 1)$. Ta lại có $(a_3, a_4) = (1, 4) \to (2, 2)$ và tiếp tục một chuỗi như sau

$$(a_2, a_3) = (1, 2) \to (1, 1)$$
$$(a_3, a_4) = (1, 2) \to (1, 1)$$

Vậy là coi như ta đã chia được 4 con gấu đầu tiên, mỗi con gấu còn đúng 1 quả dâu. Thuật toán của ta ở đây là gì?

Quá trình chia dâu sẽ diễn ra như sau : Đầu tiên, xét con gấu thứ a_{i+1} có 2^i quả dâu theo giả thiết đề bài. Và lúc này ta đã thực hiện thuật toán của ta tới con gấu thứ a_i và có 1 chuỗi $(a_1, a_2, ..., a_i) = (1, 1, ..., 1)$. Ta chọn cặp $(a_i, a_{i+1}) = \left(1, 2^i\right)$ và thực hiện bước chuyển

$$(a_i, a_{i+1}) = \left(1, 2^i\right) \to \left(2^{i-1}, 2^{i-1}\right)$$

sau đó lui dần về sau, chọn cặp $(a_{i-1}, a_i) = \left(1, 2^{i-1}\right) \to \left(2^{i-2}, 2^{i-2}\right)$ và cứ tiếp tục mãi như vậy cho tới cặp $(a_1, a_2) = (1, 1)$ và quay trở lại $(a_2, a_3) = (1, 2) \to (1, 1)$. Tiếp tục đẩy dần lên cho tới khi $a_{i+1} = 1$.

Cuối cùng ta sẽ duyệt qua hết toàn bộ các con gấu và thuật toán kết thúc khi mỗi con gấu còn lại đúng 1 quả dâu.

Cách trên là hoàn toàn tối ưu vì ta đã nhận xét rằng không thể có con gấu nào có số dâu bằng 0 và mỗi con gấu sẽ có ít nhất là 1 quả dâu.

Bài 3. Cho một số viên bi có khối lượng khác nhau từng đôi một và một cái cân thăng bằng. Biết rằng chiếc cân này không cho phép đo chính xác khối lượng của viên bi mà mỗi lần cân chỉ cho phép so sánh khối lượng của hai viên bi bất kì. Mục tiêu cuối cùng là có thể sắp xếp các viên bi này theo thứ tự khối lượng tăng dần bằng một số lần cân hữu hạn.

a/ Chứng minh rằng với 4 viên bi bất kì ta chỉ cần sử dụng 5 lần cân

b/ Chứng minh rằng với 2^n viên bi bất kỳ với n nguyên dương, ta chỉ cần sử dụng $S = (n-1)2^n + 1$ lần cân là đủ.

Lời giải.

a/ Cho 4 viên bi có khối lượng là a, b, c, d chẳng hạn.

Ta không biết khối lượng cụ thể của mỗi viên bi và mỗi lần cân như vậy chỉ đơn thuần là so sánh khối lượng giữa 2 viên bi. Nếu như ta duyệt từng viên bi một thì sẽ tốn rất nhiều lượt. Cho nên thay vì duyệt hết tất cả mỗi viên bi lần lượt như vậy, ta sẽ tách ra các viên bi ra thành 2 nhóm bằng nhau. Chiến thuật trên có thể hiểu nôm na là chia để trị. Thay vì làm việc trên các cấu hình quá lớn thì ta có thể chia nhỏ ra từng bài toán nhỏ để giải quyết. Bây giờ ta xét hai nhóm bi (a, b) và (c, d). Tiếp theo ta sẽ tiến hành so sánh từng cặp bi trong mỗi nhóm. Không mất tính tổng quát ta có thể giả sử kết quả sau khi thực hiện hai lần cân sẽ là $a < b, c > d$.

Ta so sánh d với a và b. Sẽ có các trường hợp sau

+Nếu như $d > b$ thì thuật toán dừng lại và có dãy $c > d > b > a$. Ở đây ta cần 3 lần cân

+Nếu như $d < b$ thì ta tiếp tục so sánh c và b. Nếu như $c > b$ thì ta có sắp xếp như sau $c > b > d$. Lúc này ta cần xác định vị trí của a. So sánh d và a là xong

Còn ngược lại nếu như $c < b$ thì ta có sắp xếp $b > c > d$. Và tương tự ta cũng cần xác định vị trí của a. Đầu tiên ta so sánh a và c. Nếu như $a > c$ thì ổn, còn nếu như $c > a$ thì ta lại so sánh a và d thì xong. Vậy trường hợp tệ nhất này cần đúng 5 lần cân.

b/ Ý tưởng rõ ràng nhất đó chính là quy nạp

Trước khi làm câu b/ ta sẽ sắp xếp lại một chút thuật toán mà ta đã làm ở câu a/ cho trường hợp 4 viên bi. Đầu tiên ta chia số bi hiện đang có thành 2 nhóm bằng nhau. Sau đó tiến hành sắp xếp số bi ở 2 nhóm theo thứ tự tăng dần. Tiếp theo đó ta tiến hành so sánh số bi ở hai nhóm. Duyệt từ viên bi có khối lượng nhỏ nhất của nhóm này so với viên bi có khối lượng lớn nhất ở nhóm kia. Rồi từ từ giảm dần xuống. Cách này khá hiệu quả vì thực hiện so sánh như vậy ta có thể gộp cả 2 nhóm lại với nhau, theo sau viên bi có khối lượng nhỏ nhất. Và các viên bi còn lại ta cũng so sánh liên tục từ nhỏ đến cao rồi sắp xếp chúng lại với nhau.

Phần còn lại là tính toán một chút.

Giả sử quy nạp, với 2^n viên bi ta cần sử dụng $(n-1)2^n + 1$ lần cân. Lúc này xét 2^{n+1} viên bi và mục tiêu là chứng minh số lượt cân tối đa sẽ là $n2^{n+1} + 1$.

Bắt đầu thực hiện thuật toán như sau : Đầu tiên chia 2^{n+1} viên bi thành 2 phần bằng nhau gồm 2^n viên bi. Lúc này theo giả thiết quy nạp thì số lượt cân cần thực hiện để sắp xếp các viên bi ở hai nhóm sẽ là $(n-1)2^n + 1$ ở mỗi nhóm. Tổng cộng sẽ có tối đa

$$(n-1)2^{n+1} + 2$$

lần cân. Bây giờ ta sẽ so sánh từng viên bi trong mỗi nhóm 2^n viên bi ở trên để tiến hành sắp xếp. Ta gọi 2 nhóm bi trên như sau : nhóm 1 gồm $a_1 > a_2 > ... > a_{2^n}$ và nhóm 2 gồm $b_1 < b_2 < ... < b_{2^n}$. Đầu tiên chọn ra viên bi a_{2^n} và so sánh lần lượt với 2^n viên bi của nhóm 2. Trường hợp tệ nhất là ta có $a_{2^n} < b_2 < b_3 < ... < b_{2^n}$ và ta phải thực hiện bước so sánh cuối cùng giữa a_{2^n} và b_1. Tương tự với viên bi a_{2^n-1} thì trường hợp tệ nhất đó chính là ta buộc phải so sánh cặp (a_{2^n-1}, b_2). Cứ như vậy, số lần cân tối đa, cũng là số lượt cân tệ nhất có thể xảy ra sẽ là $1+2+...+2^n = 2^{n+1}-1$. Và tổng cộng sẽ cần tối đa $(n-1)2^{n+1}+2+2^{n+1}-1 = n2^{n+1}+1$. Đây cũng là điều phải chứng minh.

Bài 4. Trên một đường tròn có đặt các điểm $A_1, A_2, ..., A_n$ theo thứ tự đó với $n > 3$ và điểm O là tâm đường tròn này. Người ta có đặt hữu hạn các lá bài vào các điểm này theo thứ tự bất kì. Mỗi lượt, chúng ta chọn thực hiện một trong các hành động sau

i/ Nếu có nhiều hơn hai lá bài ở điểm A_i, chúng ta có thể lấy 3 lá bài từ điểm A_i và đặt mỗi lá vào hai điểm bên cạnh là A_{i-1} và A_{i+1} và điểm O. Trong đó điểm A_1 và A_n là hai điểm cạnh nhau

ii/ Nếu có ít nhất n lá bài ở điểm O, chúng ta có thể lấy n là bài từ điểm O và đặt mỗi lá vào các điểm $A_1, A_2, ..., A_n$.

Chứng minh rằng nếu người ta đã đặt ít nhất $n^2 + 3n + 1$ lá bài, thì chúng ta có thể thực hiện các hành động trên một số lần sao cho mỗi điểm có ít nhất $n + 1$ lá bài.

China MO 2010

Lời giải.

Bài toán dường như có quá nhiều trường hợp bởi lẽ các lá bài được đặt một cách ngẫu nhiên vào $n + 1$ điểm $A_1, A_2, ..., A_n$ và O.

Để ý rằng, ta có thể thực hiện thao tác i/ một cách liên tục, cho điểm A_i bất kì mà có nhiều hơn 2 lá bài. Ta sẽ làm như vậy cho tới khi không còn điểm A_i nào có nhiều hơn 2 lá bài nữa.

Thế thì, ta sẽ chuyển về xử lí bài toán trong tình huống mỗi điểm A_i như vậy có 0,1 hoặc 2 lá bài.

Ban đầu ta có tổng cộng n^2+3n+1 lá bài và khi thực hiện hành động i/ cho tới khi đưa về trạng thái trên thì điểm O có ít nhất $n^2+3n+1-2n=n^2+n+1$ lá bài. Vì điểm O có số là bài lớn hơn n cho nên ta có thể thực hiện hành động số ii/. Cụ thể ta sẽ lấy n lá bài và chia đều cho các điểm A_i thì khi đó mỗi điểm sẽ có 1,2 hoặc 3 lá bài.

Trong số các điểm A_i, giả sử có x điểm chứa 1 lá bài, y điểm chứa 2 lá bài và z điểm chứa 3 lá bài với $x+y+z=n$. Do đó số lá bài ở điểm O là

$$n^2+3n+1-x-2y-3z=n^2+n+1+x-z$$

Nếu như $x \geqslant z$ thì

$$n^2+n+1+x-z \geqslant n^2+n+1$$

và ta thực hiện được n lần thao tác ii/ chuyển n^2 là bài từ O đều sang các điểm A_i và khi đó mỗi điểm có ít nhất $n+1$ lá bài. Như vậy chúng ta chỉ cần thực hiện các hành động sao cho số điểm A_i chứa 1 lá bài nhiều hơn số điểm A_i chứa 3 lá bài. Nói cách khác, chúng ta phải xuất phát tại trạng thái mà số điểm A_i chứa 1 lá bài nhiều hơn số điểm A_i chứa 3 lá bài.

Vấn đề đặt ra ở đây là : làm thế nào để chuyển các lá bài như vậy. Thực ra ta chỉ cần suy nghĩ đơn giản như sau : xét một chuỗi các điểm A_i mà chứa 3 lá bài ví dụ như sau $(a,3,3,...,b)$ trong đó a,b khác 3. Sau đó ta sẽ thực hiện hành động i/ liên tiếp để lan tỏa dần các lá bài ra. Bắt đầu từ điểm có 3 lá bài đầu tiên ở bên trái, chuyển 3 lá từ điểm này sang 2 điểm bên cạnh và sang O.

$$\begin{aligned}(a,3,3,...,b) &\to (a+1,0,4,...,b) \to (a+1,1,1,4,...,b)\\ &\to (a+1,1,2,1,4,...,b)\end{aligned}$$

và cứ như vậy ta được chuỗi $(a+1,1,2,...,1,b+1)$ thì chuỗi này không còn điểm nào chứa 3 lá bài nữa (ta tạm chưa xét $a+1,b+1$). Như vậy, số các điểm chứa 1,2,3 lá bài đều không giảm và mỗi lần thực hiện thao tác như vậy thì số lá bài ở mỗi điểm A_i sẽ giảm đi 1 cho nên ta không thể thực hiện hành động i/ vô hạn lần nữa. Ta sẽ tiếp tục chuyển các lá bài như trên cho tới khi trên đường tròn không còn chuỗi 3 lá bài nào khác nữa.

Bây giờ, giả sử rằng dù ta chuyển như thế nào đi chăng nữa thì số điểm A_i chứa 3 lá bài luôn nhiều hơn số điểm A_i chứa 1 lá bài. Như vậy, phải tồn tại hai điểm A chứa 3 lá bài mà không tồn tại điểm nào có chứa 1 lá bài nằm giữa chúng. Chúng ta có chuỗi điểm như sau

$$(a,3,2,...,2,3,b)$$

với a,b khác 3. Với chuỗi như vậy, ta thực hiện hành động i/ lần lượt từ điểm có 3 lá bài ở bên trái cho tới hết chuỗi. Thì chuỗi sẽ đưa về $(a+1,1,...,1,b+1)$. Lúc này, số lượng các cặp điểm A chứa 3 lá bài mà ở giữa chúng không chứa điểm nào chứa 1 lá bài giảm xuống. Cứ tiếp tục dần dần như vậy, ta sẽ có được điều cần chứng minh.

Bài 5. Trên một lưới ô vuông vô hạn, ta đặt một hữu hạn số xe ô tô trên mỗi ô vuông. Mỗi xe chỉ chiếm đúng một ô và không có xe nào ở chung 1 ô với nhau, đồng thời mỗi xe hướng về một trong bốn hướng cơ bản đó là lên, xuống, sang trái hoặc sang phải. Biết rằng, ở ngay trước mỗi xe đều có một ô trống và không có 2 xe nào hướng mặt vào nhau. Ở mỗi lượt đi, ta sẽ chọn một xe ô tô và di chuyển nó tiến lên trước vào ô trống trước mặt. Chứng minh rằng, ta có thể thực hiện một chuỗi các bước đi sao cho mỗi xe được đi vô số lần.

USA TSTST 2019

Lời giải.

Ta cần thực hiện bước đầu tiên để kiểm soát các xe đang có trên lưới ô vuông hiện tại. Trước hết, vì số xe trên lưới là hữu hạn cho nên tồn tại một hình chữ nhật sẽ bao phủ toàn bộ các xe ở trên.

$\uparrow$ $\triangle$						
						$\triangleright \rightarrow$
$\leftarrow \triangleleft$				$\uparrow$ $\triangle$		$\triangleright \rightarrow$
$\bigtriangledown$ $\downarrow$			$\bigtriangledown$ $\downarrow$			

Ta sẽ di chuyển dần từ từ các xe trên ra khỏi bảng và cũng nhận xét rằng chỉ cần các xe thoát ra được khỏi bảng thì chúng có thể đi được vô số lần mà không bị chặn lại. Ta chọn ra các xe mà trên đường đi của chúng ra khỏi bảng không có xe nào chặn ở trước, sau đó sẽ cho các sẽ này di chuyển ra khỏi bảng. Tiếp đó ta chỉ cần xét một bảng khác sau khi loại đi các xe như trên đi. Ví dụ một bảng như dưới đây

	$\triangleright$		$\bigtriangledown$			
	$\triangleright$			$\bigtriangledown$	$\bigtriangledown$	
$\triangleright$		$\triangle$	$\triangleright$			
	$\triangle$		$\triangleleft$		$\triangleleft$	

Bây giờ, ta sẽ sử dụng giả thiết, mỗi xe đều có một ô trống ở trước mặt. Thế thì ta xét một hàng bất kì, chẳng hạn là hàng thứ 2 tư trên xuống như bảng trên. Ta sẽ cho các loại xe di chuyển dọc dịch chuyển về ô vuông ở trước mặt. Bằng cách này, hàng sẽ chỉ còn lại các sẽ di chuyển ngang và con đường đã được giải phóng hoàn toàn cho các xe này di chuyển. Lưu ý rằng giả thiết đề bài cũng nói rằng không có hai xe nào đối mặt theo hướng ngược lại cho nên trên mỗi hàng hoặc chỉ có các xe di chuyển sang trái hoặc sang phải. Vậy ta sẽ cho các xe ngang ở hàng trên di chuyển liên tục cho tới khi ra khỏi bảng và giải phóng tất cả các xe ra khỏi hàng này.

	$\triangleright$		$\bigtriangledown$			
	$\triangleright$	$\rightarrow$	$\rightarrow$	$\downarrow$	$\downarrow$	
$\triangleright$		$\triangle$	$\triangleright$	$\bigtriangledown$	$\bigtriangledown$	
	$\triangle$		$\triangleleft$		$\triangleleft$	

Chiến thuật của ta là làm bốc hơi hết toàn bộ các xe di chuyển ngang ra khỏi bảng để chỉ còn lại các xe di chuyển dọc. Vậy ta cần tìm một thuật toán tối ưu hơn để không bị rơi vào ngõ cụt. Ta sẽ đi từ từ từng hàng một.

Xét hàng đầu tiên của bảng tính từ trên xuống. Đầu tiên, ta cho các xe di chuyển dọc dịch chuyển 1 ô về phía trước rồi sau đó cho các xe dịch chuyển ngang đi ra khỏi bảng. Tiếp theo sau đó ta sẽ xét hàng thứ ba thay vì xét hàng thứ 2. Lý do giải thích cho việc chọn hàng thứ 3 vì ta vẫn có khả năng rơi vào trường hợp tệ nhất đó chính là trường hợp sau khi các xe ở hàng thứ nhất đi xuống hàng thứ hai thì vô tình chặn mất đường đi của các xe hàng thứ 2 cho nên ta cần nghĩ cách để di dời các xe ở hàng 2 trước. Muốn làm được điều đó thì ta sẽ sử dụng hàng thứ 3.

Ở hàng thứ 3, ta xét các xe đi theo chiều dọc và cho các xe này dịch chuyển 1 bước tiến lên ô trước mặt. Nếu là các xe hướng lên trên thì coi như hoàn toàn được giải phóng vì lúc này hàng 1 cũng đã mở. Còn lại các xe hướng xuống dưới. Do đó sau khi di chuyển trên hàng 2 chỉ còn lại các xe hướng xuống dưới và ta sẽ cho các xe chạy ngang ở hàng 2 di dời ra khỏi bảng hoàn toàn. Lúc này hàng 1 và hàng 2 đã được giải phóng. Bây giờ vấn đề còn lại là hàng 3. Và ta làm tương tự y hệt như trên. Xét hàng 4, ta sẽ dịch chuyển các xe hướng lên trên ra khỏi bảng, và chỉ còn lại các xe hướng xuống dưới. Dịch chuyển các xe hướng xuống dưới này thêm 1 ô trước mặt, sau đó cho các sẽ đi ngang rời ra khỏi bảng. Tiếp theo ta chỉ cần cho các xe hướng xuống dưới ở hàng 3 đi về ô trước mặt, vì lúc này hàng 4 đã trống hoàn toàn. Sau đó ta lại cho các sẽ đi ngang ở hàng 3 rời khỏi bảng. Và cứ tiếp tục như vậy, ta sẽ từ từ giải phóng được toàn bộ tất cả các xe. Cuối cùng ta mô tả thuật toán lại như sau : để thuận tiện, ta sẽ tô màu xen kẽ các hàng của bảng bởi màu xanh, đỏ. Thuật toán sẽ diễn ra như sau, hàng đầu tiên sẽ là hàng màu đỏ

i/ Đầu tiên, cho các xe ở hàng đỏ mà nằm dọc, đi về một ô ở trước mặt. Sau đó ta sẽ cho các xe nằm ngang ở hàng này đi ra khỏi bảng

ii/ Tiếp theo ta sẽ xét hàng màu đỏ tiếp theo sau hàng màu đỏ vừa rồi. Cho các xe đi dọc dịch chuyển về 1 ô trước mặt. Luôn có cách để di chuyển các xe này theo giả thiết của bài toán. Sau đó cho các xe ngang đi khỏi bảng. Lúc này hàng màu đỏ hoàn toàn trống

iii/ Sau đó, ở hàng màu xanh nằm giữa hai hàng màu đỏ này, ta sẽ cho các xe dọc đi về các ô trước mặt vì hai hàng màu đỏ đều hoàn toàn trống nên ta di dời được. Sau đó chỉ cần cho các xe nằm ngang rời khỏi hàng xanh là xong.

Tiếp tục quá trình này cho tới khi không còn xe nào trên bảng và bài toán hoàn tất.

Bài 6. Cho $n \geqslant 2$ là một số nguyên dương. Xét trên mặt phẳng, n đoạn thẳng đôi một cắt nhau và không có 3 đoạn thẳng nào đồng quy. Mỗi đoạn thẳng như vậy sẽ gồm 2 đầu nút ở 2 phía và $n-1$ nút giao điểm là giao điểm của nó với $n-1$ đoạn thẳng còn lại. Thầy Minh chơi một trò chơi sau trên n đoạn thẳng này : đầu tiên, thầy đặt một con ốc sên vào một trong hai đầu nút của các đoạn thẳng trên, tổng cộng sẽ có n con ốc sên. Ở mỗi lượt, sau khi thầy vỗ tay thì các con ốc sên sẽ tiến lên phía trước và đứng vào giao điểm ở trước mặt. Bởi vì có $n-1$ giao điểm tất cả cho nên các con ốc sên sẽ đi qua hết toàn bộ giao điểm sau $n-1$ lần vỗ tay (lúc đầu các con ốc sên đứng ở vị trí đầu nút cho nên mất $n-1$ bước mới đi được tới nút giao cuối cùng).

a) Chứng minh rằng với n lẻ thì thầy Minh có cách đặt các con ốc sên sao cho không có hai con nào gặp nhau tại cùng một giao điểm.

b) Chứng minh rằng với n chẵn thì cho dù thầy Minh đặt các con ốc sên như thế nào thì cũng luôn có một thời điểm mà tại đó có 2 con ốc sên giao nhau tại cùng một giao điểm.

IMO SL 2016

Lời giải.

Để thuận tiện ta sẽ xét các ký hiệu sau: (s_i, l_i) trong đó s_i là con ốc sên trên đoạn thẳng l_i và A_i, B_i là hai đầu mút của đoạn thẳng

Việc đầu tiên ta cần làm đó là hãy thử hình dung ra mô hình n đoạn thẳng đôi một giao nhau ở trên có đặc điểm gì. Đồng thời ta cũng cần tìm cách để đánh số các đoạn thẳng trên để thuận tiện cho việc lập luận.

Xét trường hợp n lẻ trước.

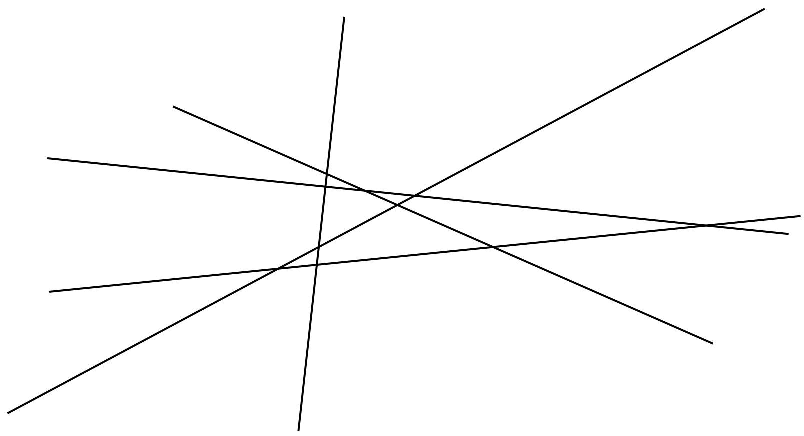
Ta chọn ra một đoạn thẳng bất kì và xét $n-1$ đoạn thẳng còn lại. Đoạn thẳng mà ta chọn chia mặt phẳng thành 2 phần và mỗi phân như vậy sẽ chứa các đầu nút của các đoạn thẳng ở trên. Tổng cộng có $2n-2$ đầu mút cho $n-1$ đoạn thẳng. Tiếp theo ta sẽ đi đến một nhận xét gần như là mấu chốt của bài toán : xét 2 đoạn thẳng bất kì và ta sẽ chọn 2 đầu nút nằm cùng phía có bờ là đoạn thẳng được chọn của 2 đoạn thẳng này. Ta tạm gọi đoạn thẳng được chọn ban đầu là l. Hai đường thẳng mà ta chọn sẽ là l_i, l_j. Đường thẳng l chia mặt phẳng thành 2 miền, không mất tính tổng quát, ta sẽ xét các đầu mút nằm về bên tay phải là B_i còn các đầu mút nằm về bên tay trái sẽ là A_i.

Đối với hai đoạn thẳng l_i, l_j ta sẽ chọn 2 đầu mút là A_i, A_j nằm cùng phía so với l. Nếu như hai con ốc sên bằng đầu di chuyển từ hai đầu mút này thì cả hai sẽ cùng gặp nhau tại một điểm và điểm này chính là giao của hai đường thẳng l_i, l_j, tạm gọi là điểm P. Lý giải cho quan sát này như sau : bắt đầu từ nút giao đầu tiên của cả hai đoạn l_i, l_j cho tới giao điểm P, thì mỗi đường thẳng mà cắt l_i cũng sẽ đồng thời cắt l_j theo nhận xét 2 đường đôi một giao nhau ban đầu. Suy ra khoảng cách từ điểm nút A_i và A_j cho đến P sẽ bằng nhau : $d(A_i, P) = d(A_j, P)$.

Do đó xét đường thẳng l, ta sắp xếp mặt phẳng thành 2 phần. Ý tưởng là dồn hết về một phía rồi từ từ di chuyển sau.

Đến đây ta lại xét thêm một quan sát nữa như sau : Xét hai đường thẳng l_h, l_k có hai con ốc sên giao nhau tại H thì ta chỉ cần đảo vị trí xuất phát của 1 trong 2 con ốc sên s_h hoặc s_k thì sau khi vỗ tay lại, 2 con này không giao nhau nữa. Thật vậy, vì $n-1$ là số chẵn cho nên không tồn tại điểm nút nào ở giữa mà cách đều 2 đầu nút cả. Trừ khi $n-1$ là số lẻ thì mới tồn tại. Do đó không bao giờ ta rơi vào trường hợp 2 đoạn l_h, l_k giao nhau tại vị trí chính giữa của mỗi đoạn.

Bây giờ quay lại cấu hình mà ta đang xét ở trên. Hình vẽ tượng trưng như dưới đây :



Đoạn thẳng l mà ta chọn sẽ bao gồm $n-1$ điểm nút và 2 đầu nút là A và B. Coi $n-1$ điểm nút đó là $h_1, h_2, ..., h_{n-1}$. Xét $n-1$ đoạn thẳng $l_1, l_2, ..., l_{n-1}$ đi qua các giao điểm này và đánh số chúng, bắt đầu từ đoạn thẳng đi qua h_1 thì sẽ đánh số là l_1 và có hai đầu mút là (A_1, B_1). Sau đó làm tương tự.

Tiếp theo ta sẽ đặt tất cả $n-1$ con ốc sên ban đầu ở $A_1, A_2, ..., A_{n-1}$. Còn con ốc sên ở l ta đặt tại A. Đầu mút A ta chọn sẽ tùy vào thế hình đang có, ta sẽ chọn A sao cho sau một vài lần vỗ tay thì s và s_1 sẽ gặp nhau tại h_1, s và s_2 sẽ gặp nhau tại h_2 và tiếp tục như vậy nhưng chỉ tới được s và $s_{\frac{n-1}{2}}$ gặp nhau tại $h_{\frac{n-1}{2}}$. Ta tưởng tự trên đoạn thẳng l sẽ gồm 2 "thanh chắn" tại hai vị trí là $h_{\frac{n-1}{2}}$ và $h_{\frac{n+1}{2}}$. Thế thì như nhận xét ở trên ta chỉ cần đảo vị trí của A thành B thì sau khi vỗ tay lại s và s_1 sẽ không còn gặp nhau nữa. Từ s đến $s_{\frac{n-1}{2}}$ cũng sẽ không gặp nhau. Còn lại là $s_{\frac{n+1}{2}}, ..., s_{n-1}$ sẽ gặp s tại $h_{\frac{n+1}{2}}, ..., h_{n-1}$. Để giải quyết thì tương tự như trên ta đảo vị trí xuất phát của các con ốc sên từ $A_{\frac{n+1}{2}}, ..., A_{n-1}$ thành $B_{\frac{n+1}{2}}, .., B_{n-1}$. Lúc này ta chỉ cần chứng minh với 2 con ốc sên với vị trí xuất phát là A_i, B_j bất kì khác phía so với đường thẳng l thì chúng sẽ không bao giờ gặp nhau. Thật vậy, giả sử chúng gặp nhau tại một điểm P chẳng hạn. Hai con ốc sên sẽ di chuyển từ 2 chiều khác nhau (so với l). Nhưng mặt khác tính từ A_i, A_j đến P thì số giao điểm là bằng nhau, tính từ A_i, B_j đến P thì số giao điểm cũng bằng nhau và cũng bằng chính cho khoảng cách từ A_j đến P. Do đó chỉ xét riêng trên đoạn l_j thì số các nút giao điểm sẽ bao gồm 2 lần số nút giao điểm từ A_j đến P cộng thêm với P là một số lẻ. Cho nên điều này mâu thuẫn với giả thiết $n-1$ là số chẵn. Vậy ta có điều vô lí.

Trường hợp 2 điều thuộc về cùng một miền không thể cắt nhau là hiển nhiên vì như ta đã nhận xét ở trên do khoảng cách từ các A_i đến đường thẳng l là khác nhau cho nên giả sử nếu như có hai điểm A_i, A_j cùng thuộc về một bờ so với l. Gọi P là giao điểm của hai đoạn l_i, l_j thì lúc này P phải nằm khác phía so với A_i, A_j như đã nói ở trên. Do P nằm khác phía và phía bên kia các điểm B_i, B_j cũng cách đều P cho nên lập luận tương tự, số giao điểm sẽ là số lẻ và ta có điều mâu thuẫn.

Lưu ý ở trên, các điểm $A_1, A_2, ..., A_{\frac{n-1}{2}}$ không nhất thiết phải theo thứ tự theo chiều kim đồng hồ trong mặt phẳng mà ta chỉ đang chỉ ra các điểm mà sau một số lần vỗ tay, thì sau khi s_i xuất phát ở A_i sẽ gặp s tại đúng h_i mà thôi.

Và đến đây ta giải quyết được trường hợp n lẻ.

Tiếp theo là trường hợp n chẵn.

Ở trường hợp này, thì từ tư tưởng xét một đường thẳng gốc như ở trên, ta sẽ suy ra rằng phải có hai đầu mút A_i, A_j nằm cùng về một phía so với l. Lúc này ta sẽ xét đoạn thẳng A_iB_i, A_jB_j và gọi P là giao điểm của hai đường thẳng này. Vì mỗi đoạn thẳng giao A_iB_i cũng sẽ giao với A_jB_j cho nên khoảng cách từ A_i tới P bằng khoảng cách từ A_j tới P và ta có điều phải chứng minh.

Nhận xét : Một bài toán khá thú vị. Tuy phần lời giải có hơi rườm ra nhưng thực chất ý tưởng không quá khó. Điểm mấu chốt là ta cần nắm bắt được tính chất của cấu hình và từ đó thực hiện quá trình "tìm và đặt" các con ốc sên vào đúng vị trí của nó.

Bài 7. Tìm số nguyên dương N lớn nhất thỏa mãn tồn tại một bảng ô vuông gồm 100 cột và N hàng thỏa mãn đồng thời 2 điều kiện sau

i/ Mỗi hàng chứa các số $1, 2, ..., 100$ theo một thứ tự nào đó

ii/ Với mỗi hàng r, s phân biệt bất kì, thì luôn tồn tại một cột c sao cho $|T(r,c) - T(s,c)| \geqslant 2$. Trong đó $T(r,c)$ là giá trị của số nằm ở hàng r, cột c.

IMO SL 2021

Lời giải.

Trước khi bắt đầu, ta định nghĩa lại một số khái niệm như sau :

Định nghĩa. Một hoán vị được gọi là hoán vị chẵn nếu như nó được biểu diễn dưới dạng tích của một số chẵn các phép chuyển vị. Tương tự cho hoán vị lẻ.

Ví dụ cho hoán vị 312 của 123 thì hoán vị này là hoán vị chẵn vì 312=(23)(12), tức 312 thu được từ 123 bằng lần lượt các phép chuyển vị giữa số hạng thứ 2 và thứ 3, rồi sau đó chuyển vị số hạng thứ 1 và thứ 2.

Quay trở lại bài toán.

Ta sẽ coi mỗi hàng của bảng như là một hoán vị của bộ số $1, 2, ..., 100$ và viết π_r là bộ hoán vị của hàng r.

Tiếp theo ta xét 50 cặp $s_k = (2k-1, 2k), k = 1, 2, ..., 50$. Thì mỗi hàng r như vậy là kết quả của các phép chuyển vị liên tiếp các cặp s_k như trên. Bây giờ ta sẽ xét hai hàng r, s sao cho π_s thu được từ π_r bằng cách thực hiện các phép chuyển vị các cặp s_k trong π_r. Tức là trong π_r ta sẽ chọn ra các số trong cặp s_k như vậy, rồi hoán đổi vị trí của chúng cho nhau. Bằng cách này ta sẽ có $|\pi_r(i) - \pi_s(i)| \leqslant 1$ với mọi i. Bởi lẽ, các cột không được chuyển thì sẽ giữ nguyên giá trị còn 2 số $2k-1$ và $2k$ đổi vị trí cho nhau cho nên tồn tại c để $|\pi_r(c) - \pi_s(c)| = |2k - 2k + 1| = 1$.

Đến đây ta có thể đánh giá được N như sau. Đầu tiên có tổng cộng 100! hoán vị được tạo nên từ 100 số như trên. Xét một hoán vị π bất kì. Lúc này ta sẽ loại ra tất cả các bộ hoán vị khác thu được từ π bằng cách thực hiện các phép chuyển vị cho 2 số trong các cặp s_k như trên. Có tổng cộng 2^{50} bộ như vậy vi phạm điều kiện ii/ của bài toán cho nên ta sẽ có tối đa $N \leqslant \dfrac{100!}{2^{50}}$ hàng.

Bây giờ ta sẽ chỉ ra rằng N như trên là lớn nhất có thể

Bước 1. Xây dựng 1 bảng thỏa mãn

Ý tưởng là thực hiện liên tiếp các quá trình để mở rộng bảng. Đầu tiên, với $n = 2$ thì bảng chỉ có 1 hàng và 2 cột. Ta điền 2 số 1,2 vào. Tiếp theo ta xét $n = 4$ và thêm 2 số 3,4 vào. Cứ như vậy cho tới $n = 2k$ thì ta thêm 2 số trong cặp s_k vào.

Thuật toán như sau : Ta sẽ thêm 2 số trong cặp s_k dựa vào vị trí của $2k-2$. Cụ thể ta sẽ thêm $2k-1, 2k$ sao cho chúng cùng tạo với $2k-2$ các hoán vị con có cùng tính chẵn lẻ. Ta không cần chúng đứng liền nhau mà chỉ cần dựa vào thứ tự xuất hiện của chúng trên mỗi hàng.

Ví dụ, ta đã có số 1 trên bảng và cần điền 2 số 2 và 3 vào. Lúc này ta mong muốn tạo ra các hoán vị con đều là hoán vị chẵn của 3 số trên. Cụ thể 3 hoán vị chẵn của 3 số trên lần lượt là 123,231,312. Cách sắp xếp như vậy là tối ưu vì nếu như xếp chồng 2 hoán vị chẵn và lẻ lên nhau thì ta không thể thêm bất cứ hoán vị con nào vào nữa và chỉ được tối đa 2 bộ, trong khi cách sắp xếp trên ta có thể thêm tối đa 3 bộ hoán vị. Ví dụ ta điền 123 là hoán vị chẵn cùng với 321 là hoán vị lẻ vào bảng thì không thể điền thêm bất cứ hoán vị nào khác nữa. Vì cho dù có điền bộ nào đi nữa thì cũng vi phạm điều kiện ii/ của bài toán

1	2	3
3	2	1
1	3	2

Vậy cách điền trên là hợp lí.

Ta cứ xây dựng dần dần lên như vậy là được. Cách điền sẽ tóm gọn lại như sau :

Dựa vào vị trí của $2k-2$ ta sẽ có 2 trường hợp

i/ Nếu như $2k-2$ xuất hiện sau hoặc trước vị trí của 2 ô trống trên hàng đó thì ta sẽ điền $2k-1, 2k$ vào 2 ô trống theo thứ tự đó.

ii/ Nếu như $2k-2$ xuất hiện ở ngay giữa vị trí 2 ô trống thì ta điền $2k$ trước, sau đó điền $2k-1$ sau theo chiều từ trái sang phải.

Minh họa thử 1 cách điền như sau : đầu tiên ta xét 2 số có sẵn trên bảng là 1,2 và ta giữ nguyên thứ tự xuất hiện của 2 số này trên mỗi hàng kể từ giờ về sau. Xét một bảng gồm 4 cột và 6 hàng đồng thời thỏa mãn yêu cầu bài toán. Thứ tự của bộ $\{2, 3, 4\}$ sẽ là $\{2, 3, 4\}, \{3, 4, 2\}, \{4, 2, 3\}$

1	2	3	4
1	4	2	3
1	3	4	2
4	1	2	3
3	1	4	2
3	4	1	2

Theo quy nạp, bảng gồm $2k-2$ cột sẽ có tối đa $N = \dfrac{(2k-2)!}{2^{k-1}}$ và chọn ra mỗi hàng như vậy, ta điền 2 số $2k-1, 2k$ vào 2 ô bất kì theo cách như trên thì sẽ có tổng cộng $N = \dfrac{(2k-2)!}{2^{k-1}} . \dbinom{2k}{2} = \dfrac{(2k)!}{2^k}$ hàng.

Bước 2. Chứng minh rằng bảng được xây dựng như trên thỏa mãn yêu cầu bài toán.

Từ cách xây dựng bảng như trên ta có nhận xét rằng trên mọi hàng thì ta luôn có các hoán vị con của $\{1,2\}, \{2,3,4\}, ..., \{98,99,100\}$ đều là hoán vị chẵn.

Tiếp theo ta chứng minh một bổ đề như sau :

Bổ đề. Cho π_1 và π_2 là hai hoán vị của $\{1,2,...,100\}$ sao cho $|\pi_1(i) - \pi_2(i)| \leqslant 1$ với mọi i. Khi đó tồn tại tập S gồm các cặp $(i, i+1)$ sao cho π_2 thu được từ π_1 bằng cách thực hiện các phép chuyển vị các cặp $(i, i+1)$ trong π_1.

Bổ đề trên có thể được chứng minh khá đơn giản bằng quy nạp.

Quay trở lại bài toán, giả sử trong bảng được xây dựng như trên tồn tại hai hàng π_1 và π_2 thỏa mãn $|\pi_1(c) - \pi_2(c)| \leqslant 1, \forall c$.

Từ bổ đề ta suy ra tồn tại một tập S sao cho mỗi phần tử trong S chênh lệch nhau ít nhất là 2 đơn vị và đồng thời với mỗi $j \in S$ thì π_2 thu được từ π_1 bằng cách thực hiện các phép chuyển vị các phần tử $(j, j+1)$ trong π_1.

Bây giờ gọi $r = \min S$ thì sẽ có 2 trường hợp. Trường hợp đầu tiên $r = 2k-1$ là số lẻ. Thì khi đó ta sẽ chuyển vị cặp $(2k-1, 2k)$. Nhưng để ý rằng trong π_1, π_2 thì các hoán vị con của $(2k-2, 2k-1, 2k)$ đều cùng tính chẵn lẻ (với $k = 1$ thì ta sẽ chuyển vị cặp $(1,2)$ cũng sẽ tạo ra điều vô lí). Nếu muốn tạo ra 2 hoán vị con cùng chẵn thì cần phải có $2k-3 \in S$ để thực hiện phép chuyển vị chứa $2k-2$. Nhưng điều này là không thể vì $r = \min S$. Cho nên trường hợp này loại.

Tiếp theo ta xét $r = 2k$ thì tương tự như vậy π_1, π_2 cảm sinh 2 hoán vị con của $\{2k, 2k+1, 2k+2\}$ cùng tính chẵn lẻ cho nên cần có $2k+2 \in S$ và tương tự $98 \in S$. Nhưng khi đó ta lại có điều mâu thuẫn vì π_1, π_2 cảm sinh hai hoán vị con khác tính chẵn lẻ của $\{98, 99, 100\}$. Và từ đây ta có điều phải chứng minh.

Bài 8. Cho trước các số nguyên dương a, b, c. Trên đường tròn có $a+b+c$ con vịt ngồi xen kẽ nhau. Các con vịt sẽ cùng nhau chơi oẳn tù tì, trong đó có a con vịt ra búa, b con vịt ra bao và c con vịt ra kéo. Mỗi lượt chơi sẽ gồm 2 con vịt chơi với nhau và diễn ra như sau

- Nếu một con vịt ra búa ngồi sau con vịt ra kéo thì chúng đổi chỗ cho nhau
- Nếu một con vịt ra kéo ngồi sau con vịt ra bao thì chúng đổi chỗ cho nhau
- Nếu một con vịt ra bao ngồi sau con vịt ra búa thì chúng đổi chỗ nhau nhau

Tóm lại nếu con vịt ngồi sau thắng con ngồi ở trước thì 2 con vịt sẽ đổi chỗ cho nhau. Tính số lượt chơi tối đa có thể xét trên mọi cấu hình theo a, b, c.

Lời giải.

Để thuận tiện ta có thể kí hiệu mỗi con vịt ra búa là R, ra kéo là S, còn ra bao là P.

Nhận xét rằng, sau khi 2 con vịt đổi chỗ cho nhau thì con vịt ngồi ở trước không thể đi lùi về sau được nữa mà chỉ có thể tiến về phía trước. Ta sẽ hình dung thao tác trên đối với một con vịt ra búa như sau : búa ngồi sau kéo cho nên bước tiếp theo búa sẽ vòng ra đứng trước kéo

$$...RS... \to ...SR...$$

Và lúc này con vịt ra búa không thể lùi về sau được nữa. Tiếp theo để tối ưu ta có thể chọn một dãy các con vịt ra búa và ra kéo đứng liền kề nhau rồi tiến hành đổi chỗ lần lượt

$$....RRRSSS... \to ...RRSRSS... \to ...RRSSRS...$$

Cứ như vậy ta sẽ đạt được số bước đi bằng tích độ dài của 2 dãy trên. Đến đây ta cũng dần mơ hồ nhận ra được trạng thái mà số bước đi đạt được tối đa chính là khi các con vịt cùng 1 loại xếp kề nhau trên đường tròn như dưới đây

$$\underbrace{RR....R}_{a}\underbrace{SS...S}_{c}\underbrace{PP...P}_{b}$$

Lúc này số bước đi tối đa sẽ là $\max\{ab, bc, ca\}$. Vấn đề là làm sao để chỉ ra rằng đây chính là con số tối ưu nhất có thể.

Thế thì ta cần dựa lại vào nhận xét trên : các con vịt, một khi đã tiến về phía trước thì không thể lùi về sau được nữa. Và tương tự, một khi đã lùi về sau thì không thể tiến lên phía trước được nữa. Ví dụ con ra bao ở phía sau con ra búa thì sau khi đổi chỗ 2 con này cho nhau thì búa chuyển về ở sau bao. Búa không thể thắng bao nên không có cách nào đi lên trước. Còn trường hợp bao chuyển sang đứng trước búa thì như đã nói ở trên, bao không thể lùi về sau nữa.

Cho nên với một bộ ba con vịt ra kéo búa bao bất kỳ trên đường tròn thì chỉ có thể có tối đa 1 lượt đi cho bộ ba này. Có tổng cộng abc bộ ba như vậy và ban đầu ta sẽ có tối đa abc bước đi. Mỗi lần ta thực hiện đổi chỗ hai con vịt thì sau khi đổi chỗ xong 1 trong 2 con sẽ không thể di chuyển tiếp được nữa. Ví dụ có bộ RSP. Nếu ta cho S thắng P và đảo vị trí S, P cho nhau thì sau khi đổi ta được bộ RPS. Lúc này P không thể di chuyển tiếp vì R thua P còn P thua S. Cho nên mỗi lần như vậy, sẽ có 1 con giảm đi và đến cuối cùng sẽ có tối đa $\dfrac{abc}{\min\{a, b, c\}} = \max\{ab, bc, ca\}$. Suy ra ta có điều phải chứng minh.

Một số bài tập khác giải vui

Bài 9. Một con rắn có độ dài k sẽ là một bộ bao gồm k ô vuông $(s_1, s_2, ..., s_k)$ trong một hình vuông gồm các ô vuông đơn vị $n \times n$. Các ô này đôi một phân biệt và hai ô s_i, s_{i+1} kề nhau với $i = 1, 2, ..., k-1$. Lưu ý rằng hai ô được gọi là kề nhau nếu như chúng có một cạnh chung. Nếu lúc ban đầu con rắn này chiếm k ô $(s_1, s_2, ..., s_k)$ và đầu của con rắn s_1 đang kề với ô vuông s thì con rắn có thể di chuyển lên trước tiến về ô s, đồng thời lúc này nó sẽ chiếm một dãy các ô $(s, s_1, s_2, ..., s_{k-1})$. Con rắn được gọi là đi vòng quanh nếu như ban đầu nó đang chiếm các ô $(s_1, s_2, ..., s_k)$ nhưng sau một hữu hạn các bước di chuyển thì nó lại chuyển sang chiếm các ô $(s_k, s_{k-1}, ..., s_1)$.

Liệu có tồn tại một số nguyên dương $n > 1$ sao cho ta có thể đặt một con rắn có độ dài $0.9n^2$ vào bảng $n \times n$ sao cho nó có thể đi vòng quanh được.

USA Winter TST for IMO 2019

Bài 10. Cho một đường đi U gồm các đỉnh phân biệt và có một người chơi đang bịt mắt đứng tại một trong các đỉnh của U. Mỗi đỉnh của đường đi U được đánh số từ 1 đến n không theo một thứ tự nhất định nào cả. Ở mỗi lượt chơi, quản trò sẽ cho người chơi biết rằng anh/cô ta đang đứng ở đỉnh có bậc là 1 hay là đỉnh có bậc là 2. Nếu như người chơi đang đứng ở đỉnh có bậc là 1 thì người chơi đó sẽ di chuyển lên đỉnh duy nhất kề với đỉnh bậc 1 đó. Còn nếu như người chơi đó đang đứng ở đỉnh có bậc là 2 thì anh/cô ta sẽ được phép chọn 1 trong 2 đỉnh để đi tiếp. Người chơi có thể quyết định chọn đỉnh có chỉ số nhỏ hơn hoặc thấp hơn trong 2 đỉnh ở trên. Trong suốt trò chơi, thông tin duy nhất mà người chơi nhận được sau k lượt chơi chính là bậc của k đỉnh mà người chơi đã đi qua và lựa chọn của anh/cô ấy trong mỗi lượt. Liệu có tồn tại một chiến thuật để người chơi có thể xác định được số đỉnh mà đường đi U đang có sau một hữu hạn lượt chơi hay không?

KoMaL A859

Bài 11. Cho n là một số nguyên dương. Tasty và Stacy có một chiếc vòng cổ được đính $3n$ hạt xâu, mỗi hạt xâu có một trong hai màu là xanh lục hoặc xanh lam, sao cho không có 3 hạt nào có cùng màu đứng cạnh nhau. Hai người cùng nhau chơi một trò chơi, mà tại đó hai người luân phiên nhau lần lượt bỏ 3 hạt xâu ra khỏi chiếc vòng cổ theo quy tắc như sau :

- Mỗi lần tới lượt của mình, Tasty chỉ có thể bốc 3 hạt liền kề có màu lần lượt là lục/lam/lục
- Mỗi lần tới lượt của mình, Stacy chỉ có thể bốc 3 hạt liền kề có màu lần lượt là lam/lục/lam

Họ sẽ thắng nếu như có thể bốc hết toàn bộ các hạt sau $2n$ lượt chơi. Chứng minh rằng, nếu họ có thể thắng khi Stacy đi trước thì Tasty cũng có thể đi trước và dành chiến thắng.

USA TST 2019

Bài 12. Có $n \geqslant 1$ quyển vở được đánh số từ 1 đến n và xếp thành một chồng. Zahar chơi một trò chơi như sau : Zahar chọn ngẫu nhiên một cuốn sổ được đánh chữ số k và không đứng đúng vị trí của nó là vị trí thứ k từ trên đếm xuống. Sau đó Zahar sẽ chuyển cuốn số này về vị trí đúng của nó là vị trí thứ k trong chồng vở trên. Nếu như không có cuốn sổ nào thỏa mãn yêu cầu trên thì Zahar sẽ dừng lại. Hỏi Zahar có thể sắp xếp được tất cả các cuốn sổ theo thứ tự tăng dần từ trên xuống sau 1 hữu hạn bước hay không?

Kyiv City MO 2024

Bài 13. Cho m, n là hai số nguyên dương với $m \geqslant 2$. Với mỗi cặp số nguyên dương x, y ta kí hiệu $D(x, y)$ là số nguyên dương d duy nhất thỏa mãn $m^d | x - y$ nhưng $m^{d+1} \nmid x - y$. Cho một tập S gồm các số nguyên dương và giả sử rằng có tối đa n giá trị nguyên dương phân biệt của $D(x, y)$ trong đó $x, y \in S$. Tính $\max |S|$.

CAMO 2022