Lời giải ICPC National 2021

Bài A
Arranging Marbles

Số AC (trước đóng băng): 135
AC đầu tiên: 19’ - HN.AMS.funny

Lời giải: Nguyễn Hoàng Vũ
NA.PBC Mr. Un

Bài A - Arranging Marbles

Tóm đề:

- Có $\mathbf{n^2}$ viên sỏi được tô bởi nhiều nhất **n + 1** màu khác nhau, các màu được đánh số từ 1 đến n + 1.
- Cần chia các viên sỏi vào n nhóm, mỗi nhóm **đúng n viên** sao cho trong các viên sỏi trong một nhóm được tô bởi **nhiều nhất 2 màu khác nhau**.

Bài A - Arranging Marbles

Lời giải:

- Luôn tồn tại cách chia các viên sỏi.
- Lặp lại quá trình sau n lần:
  - Nếu màu có ít sỏi nhất có ít nhất n viên sỏi, lấy n viên sỏi ra và cho vào 1 nhóm.
  - Ngược lại, lấy màu có nhiều sỏi nhất, **có thể chứng minh rằng tổng số sỏi của 2 màu chắc chắn không nhỏ hơn n** (*). Ta lấy một phần sỏi của màu lớn hơn bù vào màu nhỏ hơn và tách ra 1 nhóm.

Bài A - Arranging Marbles

Lời giải:

- Tại bước thứ i (1 <= i <= n), số màu khác nhau <= n - i + 2 và số sỏi còn lại là (n - i + 1) * n.
- Nếu tổng số sỏi hai màu được chọn bé hơn n, thì các màu còn lại có nhiều nhất n - 2 viên sỏi.
- Khi đó tổng số sỏi nhiều nhất sẽ là (n - 2) * (n - i) + n - 1 < (n - i + 1) * n, không thỏa mãn.

Bài B
Beautiful Board

Số AC (trước đóng băng): 266
AC đầu tiên: 9’ - 011

Lời giải: Nguyễn Việt Dũng
Vaccinated - HUST

Bài B - Beautiful board

Tóm đề:

- Cho một lưới ô vuông kích thước R x C, mỗi ô có chứa một chữ cái in hoa. Một thao tác thay đổi bảng có thể biến chữ cái trong 1 ô bất kỳ thành chữ cái tiếp theo hoặc chữ cái ngay trước nó trong bảng chữ cái tiếng Anh (tiếp theo của Z là A, trước của A là Z).

- Hỏi: Cần mất ít nhất bao nhiêu thao tác để mỗi hàng và mỗi cột của bảng đều tạo thành các xâu đối xứng?

Bài B - Beautiful board

Lời giải:

- Tất cả các hàng và các cột đều là các xâu đối xứng khi và chỉ khi 4 ô **(i, j), (i, C - j + 1), (R - i + 1, j), (R - i +1, C - j + 1)** có cùng chữ cái với mọi (i, j).

  \+ Mỗi một bộ **4 ô như vậy không bị ảnh hưởng bởi các thao tác trên các ô khác**.

  => Với mỗi bộ 4 ô như trên, ta tìm cách biến chúng về cùng 1 chữ cái sao cho số thao tác là ít nhất.

- Độ phức tạp: O(RC)

Bài C
Congruent Triangles

Số AC (trước đóng băng): 8
AC đầu tiên: 144’ -
HCMUS-RationalKitteNs

Lời giải: Nguyễn Việt Dũng
Vaccinated - HUST

Bài C - Congruent Triangles

Đề bài:

- Cho n điểm tọa độ nguyên trên mặt phẳng. Hai tam giác gọi là bằng nhau khi chúng có 3 cạnh và 3 góc tương ứng bằng nhau.

- Hỏi: Đếm số bộ 6 điểm (i1, i2, i3, j1, j2, j3) sao cho:

  + i1 < i2 < i3

  + i1, i2, i3, j1, j2, j3 đôi một phân biệt

  + Diện tích tam giác i1i2i3 và j1j2j3 dương

  + Tam giác i1i2i3 = Tam giác j1j2j3

Bài C - Congruent Triangles

Lời giải:

- Đếm trước trong O(n^3*log(n)) các số sau đây:

+ d0[x, y, z] = Số tam giác có 3 cạnh độ dài (x, y, z)

+ d1[i][x, y, z] = Số tam giác có 1 đỉnh là i và 3 cạnh độ dài (x, y, z)

+ d2[i, j][x, y, z] = Số tam giác có 2 đỉnh là (i, j) và 3 cạnh độ dài (x, y, z)

+ d3[i, j, k][x, y, z] = Số tam giác có 3 đỉnh là (i, j, k) và 3 cạnh độ dài (x, y, z)

Bài C - Congruent Triangles

Lời giải:

=> Sử dụng nguyên lý bao hàm loại trừ, giả sử tam giác i1i2i3 có 3 cạnh dài lần lượt là x, y, z, thì khi đó số tam giác bằng tam giác i1i2i3 mà không chứa đỉnh nào trong 3 đỉnh i1, i2, i3 là:

d0[x, y, z] - d1[i1][x, y, z] - d1[i2][x, y, z] - d1[i3][x, y, z] +

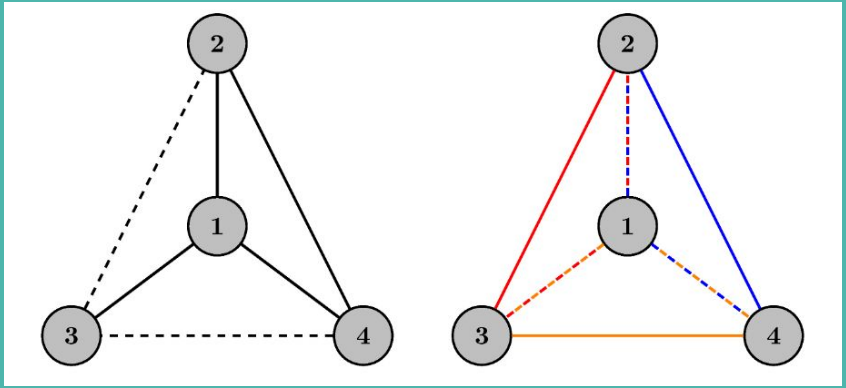
d2[i1, i2][x, y, z] + d2[i1, i3][x, y, z] + d2[i2, i3][x, y, z] - d3[i1, i2, i3][x, y, z]

- Độ phức tạp: O(n^3*log(n))

!- Lưu ý: map trong C++ chạy khá chậm cho bài này => chặt nhị phân tay trên mảng đã sắp xếp; kiểm tra 3 điểm thẳng hàng không dùng số thực; không được hash trạng thái.

Bài D
Distinctive Tours

Số AC (trước đóng băng): 1
AC đầu tiên: 199’ - PENDLE



Lời giải: Hoàng Xuân Nhật
HCMUS-RationalKitteNs

Bài D - Distinctive Tours

Tóm đề:

- Cho một đồ thị đơn (không có khuyên hay cạnh lặp), bạn cần thêm vào ít cạnh nhất sao cho đồ thị mới vẫn là đồ thị đơn, và bạn có thể chọn ra *k* chu trình đơn sao cho mỗi chu trình chứa ít nhất một cạnh **không thuộc bất cứ chu trình nào khác**.

Bài D - Distinctive Tours

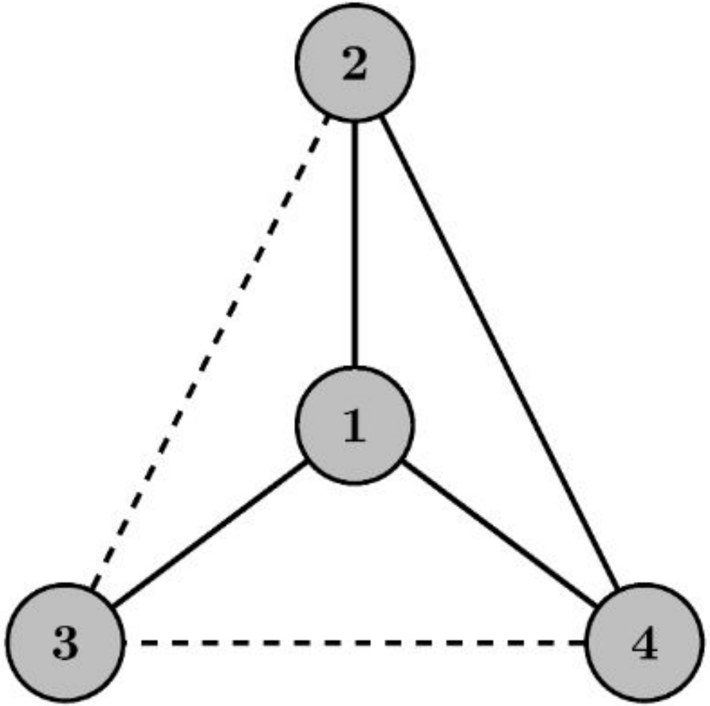
Nhận xét:

- Cho một chu trình, ta chọn cạnh (*u*, *v*) mà chỉ thuộc chu trình đó, sau đó xóa cạnh đó ra khỏi đồ thị, và tiếp tục giải với đồ thị mới.

  ⇒ Ta chỉ có thể xóa khi đồ thị còn chu trình (không phải cây)

- Xét một TPLT *n* đỉnh *m* cạnh trong đồ thị ban đầu:
  - Nếu nó là cây ⇒ không có chu trình nào cả
  - Nếu nó không phải là cây ⇒ ta có thể chọn được *m* - (*n* - 1) chu trình, bằng cách chọn các cạnh không trong cây khung

Bài D - Distinctive Tours



Bài D - Distinctive Tours

Lời giải:

- Từ nhận xét trên, ta nhận ra với mỗi hai đỉnh *(u, v*) thuộc một thành phần liên thông mà cạnh (*u*, *v*) chưa tồn tại, ta có thể thêm 1 cạnh để được thêm 1 chu trình.
- Rõ ràng ta không thể được nhiều hơn 1 chu trình với mỗi cạnh thêm vào ⇒ tối ưu
- Ta chỉ không thêm được nữa khi mọi TPLT đều là đồ thị đầy đủ
- Lúc này, ta phải gộp 2 TPLT lại

⇒ Chọn 2 TPLT lớn nhất, vì khi đó số cạnh chưa tồn tại sẽ là nhiều nhất

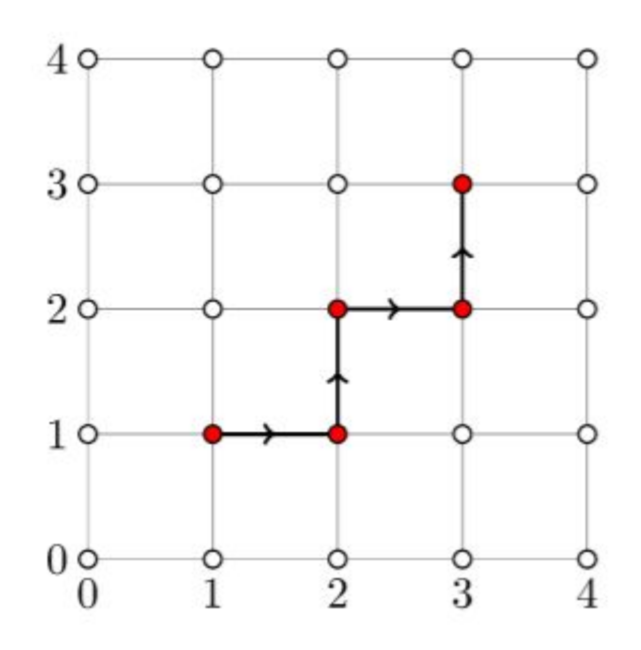
Bài D - Distinctive Tours

Đánh giá:

- Với giới hạn *n* nhỏ, ta có thể cài đặt thoải mái (ma trận kề) mà vẫn chạy đủ nhanh.

Bài E
Even Paths

Số AC (trước đóng băng): 7
AC đầu tiên: 158’ - HCMUS-RationalKitteNs



Lời giải: Lê Bảo Hiệp
HCMUS-FlamingTomatoes

Bài E - Even Paths

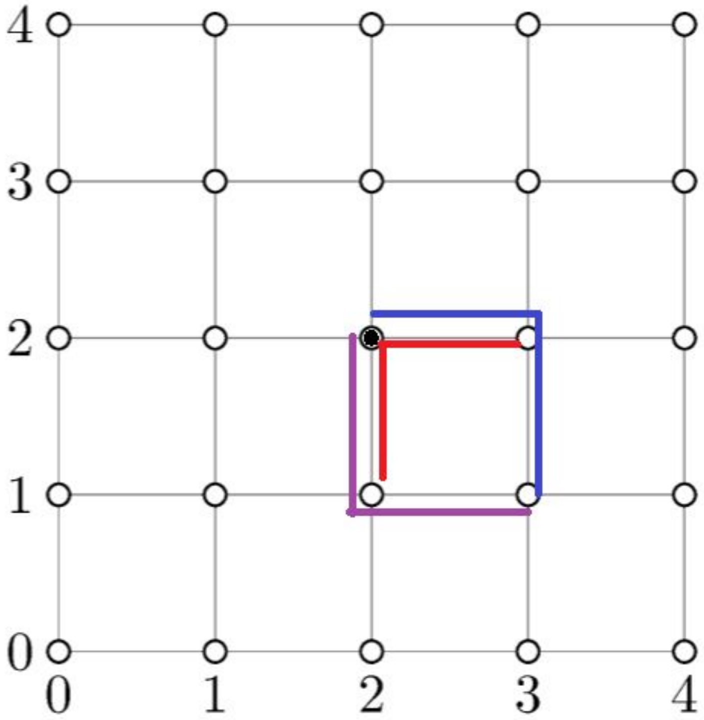
Tóm đề:

- Cho lưới ô vuông ($M + 1$) x ($M + 1$) ($M \leq 50$). Mỗi ô có một bóng đèn. Có N ô đang sáng.
- Một đường đi hợp lệ **đi qua các ô kề nhau** (4 ô kề là 4 ô trên dưới trái phải), **không có ô nào lặp lại** và **độ dài đường đi là chẵn**. Mỗi khi đi qua một ô sẽ **switch bóng đèn tại ô đó** (bật thành tắt, tắt thành bật).
- Chỉ ra các đường đi **bất kỳ** (ô xuất phát và ô kết thúc tự chọn) sao cho sau khi đi thì **tất cả bóng đèn đều tắt**.
- **Không cần tối thiểu hoá số đường đi** nhưng không vượt quá 10000.

Bài E - Even Paths

Lời giải:

- Ta *luôn có thể switch một ô bất kỳ* như sau (ví dụ với ô (2, 2)):
    - (2, 1) → (2, 2) → (3, 2)
    - (2, 2) → (2, 1) → (3, 1)
    - (2, 2) → (3, 2) → (3, 1)
- Ô (2, 2) đi qua 3 lần, các ô (2, 1), (3, 1), (3, 2) đi qua 2 lần.
- Số đường đi tổng cộng tối đa:
    - $3 * (M + 1)^2 \approx 7500 < 10000$



Bài F
Final Ranking

Số AC (trước đóng băng): 135
AC đầu tiên: 19’ - HN.AMS.funny

Rank	Contestant	Score
1	A	100
2	B	90
2	C	90
4	D	88

Lời giải: Hoàng Xuân Nhật
HCMUS-RationalKitteNs

Bài F - Final Ranking

Tóm đề:

- Trong một kì thi có *n* thí sinh, thứ hạng của một thí sinh được tính bằng số thí sinh có điểm lớn hơn thí sinh đó. Hỏi có bao nhiêu bảng xếp hạng khác nhau có thể có.
- Hai bảng xếp hạng gọi là khác nhau nếu tập các thứ hạng là khác nhau.

Bài F - Final Ranking

Lời giải:

- Quy hoạch động: gọi f[n] là số bảng xếp hạng hợp lệ với *n* thí sinh
- Trường hợp cơ bản: f[0] = 1
- Công thức quy hoạch động:

$$f(i) = \sum_{k=1}^{i} f(i-k)$$

- Ý nghĩa: ta cho *k* thí sinh cuối cùng đồng hạng, khi đó bài toán quy về bài toán con với *i* - *k* thí sinh.

Bài F - Final Ranking

Đánh giá:

- Với $n \leq 50$, lời giải trên cài đặt và chạy rất nhanh.
- Fun fact: nếu khai triển công thức ra, bạn sẽ nhận ra

$$f(n) = 2^{n-1}$$

- Giải thích: với mỗi thí sinh ngoài thí sinh đầu tiên, bạn có 2 lựa chọn: Cho thí sinh này đồng hạng với thí sinh liền trước, hoặc khác hạng.

Bài G
Group Testing

Số AC (trước đóng băng): 350
AC đầu tiên: 3' - 000

Lời giải: Lê Bảo Hiệp
HCMUS-FlamingTomatoes

Bài G - Group Testing

Tóm đề:

- Cho biết có *n* nhóm người và số người trong mỗi nhóm.
- Với mỗi nhóm người, thực hiện một test gộp. Nếu **test gộp âm tính** thì **tất cả** người trong nhóm đó **đều âm tính**. Ngược lại, nếu **test gộp dương tính** thì **cần test thêm** từng người để xác định ai dương tính.
- Cho biết trong *n* test gộp thì có chính xác *k* test dương tính.
- Tìm số test thêm *ít nhất* và *nhiều nhất* có thể.

Bài G - Group Testing

Lời giải:

- Nếu test gộp của nhóm có *x* người dương tính:
  - *x* > *1*: cần test thêm x người
  - *x* = *1*: không cần test thêm
- Sort lại các nhóm **tăng dần về số người**.
- Ít nhất: *k* nhóm **nhỏ nhất** dương tính.
- Nhiều nhất: *k* nhóm **lớn nhất** dương tính.

Bài H
Hallway Tiling

Số AC (trước đóng băng): 15
AC đầu tiên: HCMUS - Clique

Lời giải: Nguyễn Hoàng Vũ
NA.PBC Mr. Un

Bài H - Hallway Tiling

Tóm đề:

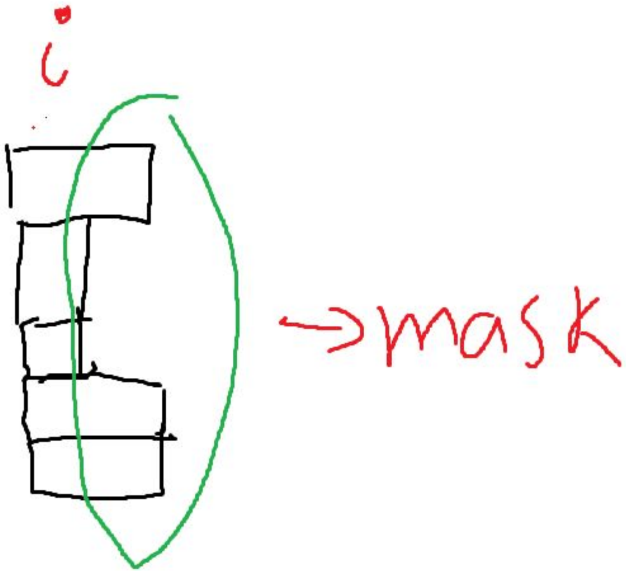
Cho một hình chữ nhật kích thước r x n, và k ô được đặt sẵn (r <= 6, n <= 10^12, k <= 10). Đếm số cách phủ kín hình chữ nhật này bởi các hình chữ nhật 1 x 2 hoặc 2 x 1.

Bài H - Hallway Tiling

Lời giải:

- Với n nhỏ, ta sử dụng dp bitmask,

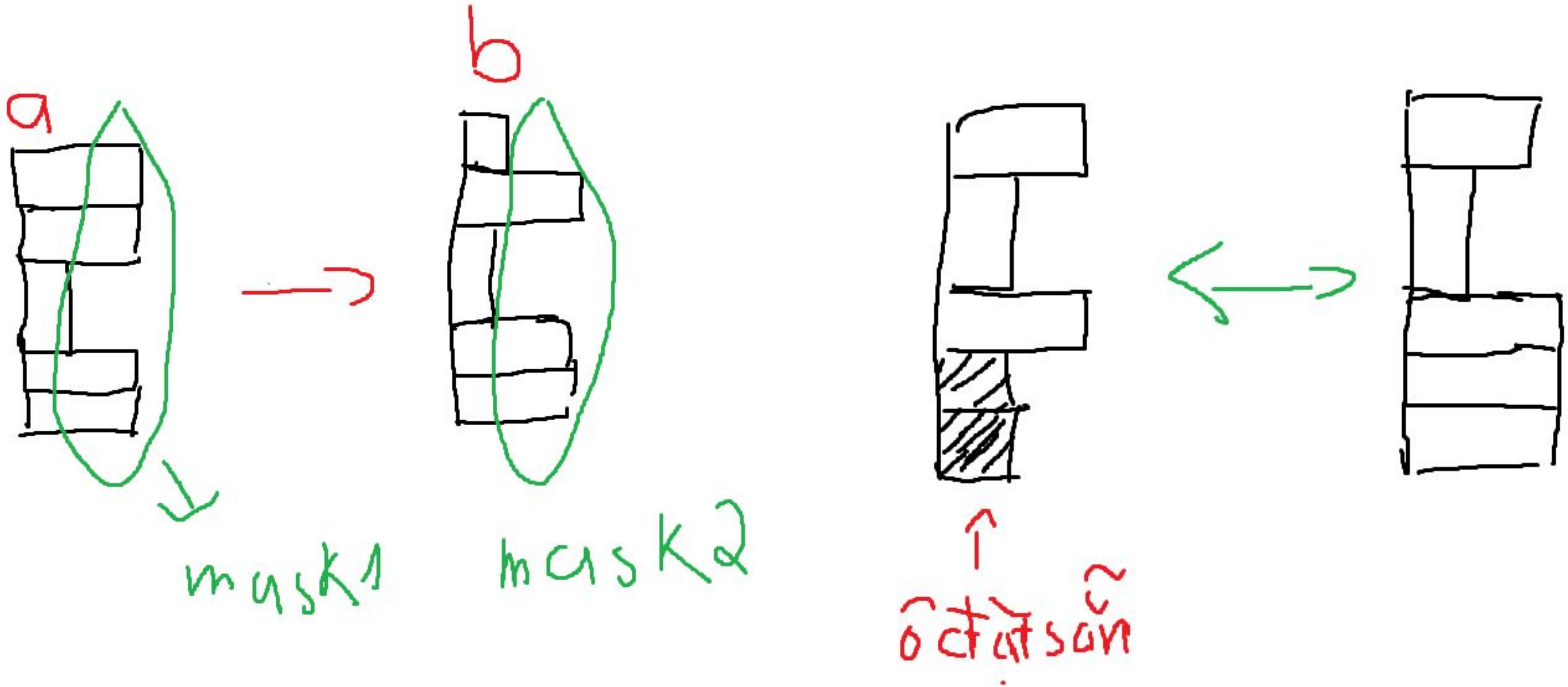
  dp[i][mask] có ý nghĩa các hcn 1x2 ở cột i và i + 1 nằm ở các bit bật có trong mask.
- Để chuyển sang i + 1 ta đặt các hcn 2x1 trước rồi đặt các hcn 1x2 sau.



Bài H - Hallway Tiling

Lời giải:

- Để làm với n lớn hơn, ta sử dụng nhân ma trận.
- Gọi các cột chứa ô đặt sẵn là y1, y2, ..., yp (y1 < y2 < ... < yp), ta chia hcn thành các khoảng (0, y1], (y1, y2], ..., (yp, n].
- Sử dụng nhân ma trận để tính với mỗi khoảng (a, b], f[mask1][mask2] là số cách để xếp các hcn nhỏ sao cho trạng thái hcn 1x2 ở a là mask1 và ở b là mask2 với giả sử cột b không có ô đặt sẵn.
- Tại cột b, để xử lý các ô được đặt sẵn, **ta coi như các ô đó được đặt các hcn 1x2**. Xét các f[mask1][mask2] mà mask2 có đủ các ô được đặt sẵn.


a
b
mask1
mask2
ô cắt sẵn

Bài H - Hallway Tiling

Lời giải:

- Gọi mask3 là mask2 sau khi tắt các bit tương ứng với các ô đặt sẵn, chuyển giá trị f[mask1][mask2] cho f[mask1][mask3], các giá trị f còn lại gán bằng 0.
- Cuối cùng nhân các ma trận f lại với nhau ta đc ma trận g, kết quả là g[0][0] (không có hcn 1x2 nằm ở m và m + 1).
- Độ phức tạp: k * (2^r)^3.

Bài I
ICPC Problem Selection

Số AC (trước đóng băng): 139
AC đầu tiên: 42’ - 011

Lời giải: Nguyễn Quang Long
HCMUS-KFCTuba

Bài I - ICPC Problem Selection

Tóm đề:

- Cho *n* bài tập (*n* ≤ 50) và 5 loại problem tags khác nhau bao gồm: dp, graph, mathgeo, ds, adhoc.
- Mỗi loại bài tập sẽ chứa một số tags trong 5 loại problem tags đã cho.
- Cho cận dưới và cận trên số lượng bài tập của mỗi loại problem tag. $(0 \le L_i \le R_i \le 15)$.
- Đếm số lượng tập con **không rỗng** trong n bài tập ban đầu, sao cho số lượng bài tập của mỗi loại problem tag thỏa điều kiện cận dưới và cận trên của loại đấy.

Bài I - ICPC Problem Selection

Lời giải:

- Quy hoạch động: gọi f[pos][cnt1][cnt2][cnt3][cnt4][cnt5] là kết quả bài toán khi xét những bài tập từ 1 đến pos, và số lượng bài của từng tag lần lượt là cnt1, cnt2, cnt3, cnt4, cnt5.
- Trạng thái cơ bản: $f_{0,\ 0,\ 0,\ 0,\ 0,\ 0} = 1$
- Chuyển dịch trạng thái:

$$f_{pos,\ cnt1,\ cnt2,\ cnt3,\ cnt4,\ cnt5} = \sum \begin{cases} f_{pos-1,\ cnt1,\ cnt2,\ cnt3,\ cnt4,\ cnt5} \\ f_{pos-1,\ ncnt1,\ ncnt2,\ ncnt3,\ ncnt4,\ ncnt5} \end{cases}$$

- Trong đó, ncnt_i = cnt_i - 1 nếu bài tập pos có problem tag loại i.
- Kết quả bài toán chính là tổng những f thỏa mãn.

Bài I - ICPC Problem Selection

Lưu ý:

- Do đề bảo chỉ đếm những tập không rỗng, nên khi cận dưới của từng loại problem tag = 0, thì ta phải trừ kết quả đi 1.
- Bài toán sẽ trở nên dễ dàng hơn nhiều nếu như bạn code bài toán bằng phương pháp đệ quy có nhớ.
- Độ phức tạp thuật toán là O(N * R^5) với R là cận trên tối đa của một loại problem tag.

Bài J
Jewel Box

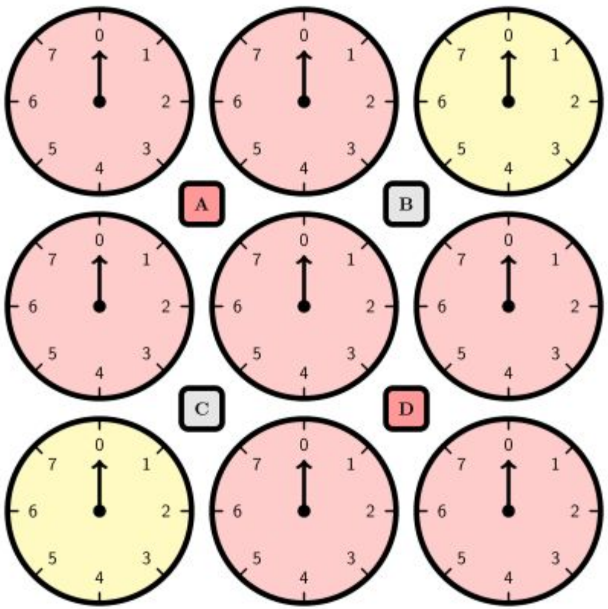
Số AC (trước đóng băng): 42
AC đầu tiên: 56’ - 010

Lời giải: Nguyễn Quang Long
HCMUS-KFCTuba

Bài J - Jewel Box

Tóm đề:

- Cho 9 ổ khóa, mỗi ổ có giá trị từ 0 đến C - 1 (C là một số cho trước).
- Bạn có thể xoay một trong bốn ổ 1, 3, 7, 9 theo chiều kim đồng hồ.
- Nếu một đồng hồ có giá trị x được xoay, giá trị của nó sẽ trở thành (x + 1) modulo C.
- Bạn đồng thời được cho bốn công tắc A, B, C, D. Khi sử dụng một công tắc, bốn ổ khóa xung quanh công tắc sẽ được kết nối. Khi đó, một ổ khóa xoay sẽ khiến những ổ khóa kết nối với nó xoay theo cùng chiều kim đồng hồ.
- Cho giá trị ban đầu của 9 ổ khóa, bạn hãy chỉ ra một cách làm thực hiện không quá 1000 thao tác để biến 9 ổ khóa đồng thời chỉ về giá trị 0.

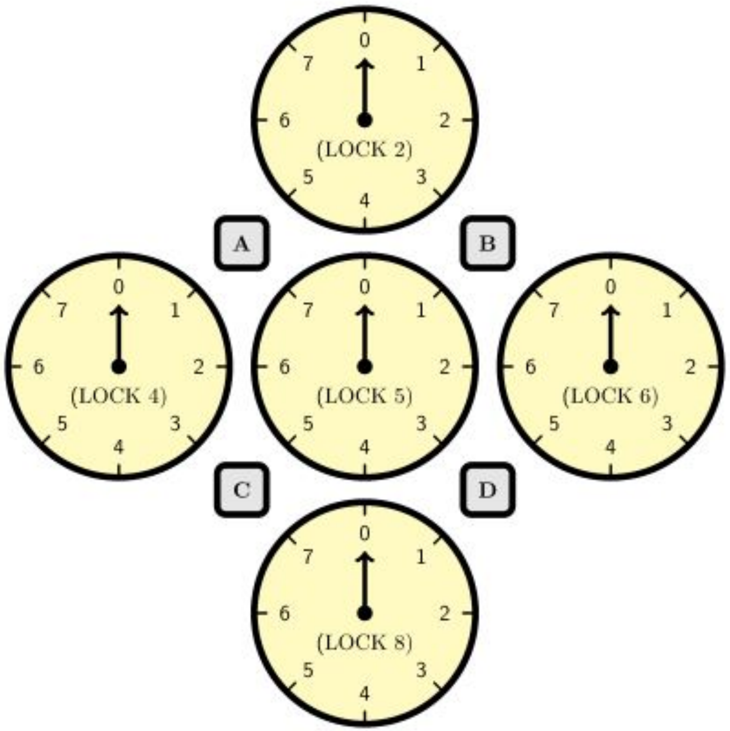


Bài J - Jewel Box

Lời giải:

- Cần xoay tất cả ổ khóa từ 1 đến 9 về cùng một giá trị X, sau đó từ X, sử dụng hết 4 công tắc để khiến 9 ổ khóa về giá trị 0.

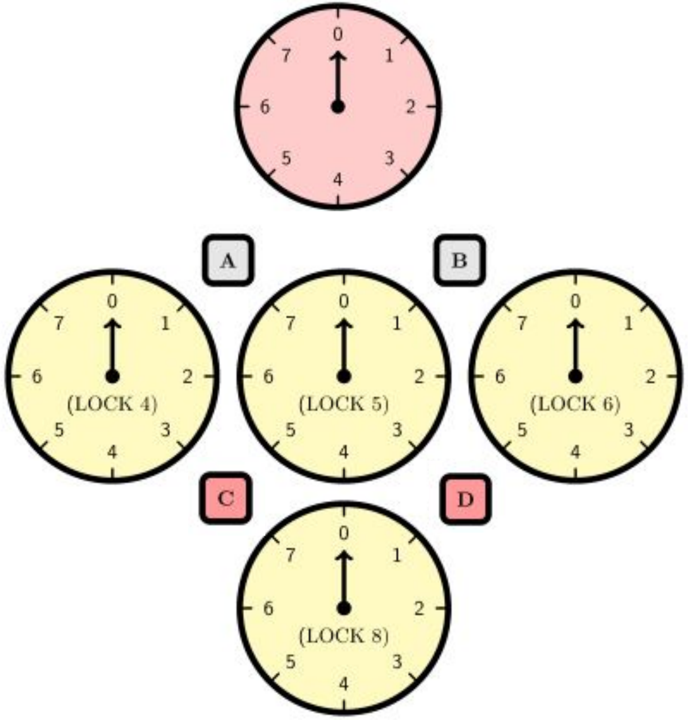
- Do ổ khóa 1, 3, 7, 9 có thể được xoay độc lập, ta chỉ cần quan tâm đến những ổ khóa 2, 4, 5, 6, 8, và thực hiện xoay các ổ khóa 1, 3, 7, 9 sau cùng.



Bài J - Jewel Box

Lời giải:

- Nhận xét: khi bật công tắc C - D và xoay, thay vì xoay các ổ khóa 4, 5, 6, 8 theo chiều kim đồng hồ, ta có thể xem như đang **xoay ổ khóa 2 ngược chiều kim đồng hồ**.
- Tương tự với các trường hợp bật công tắc B - D, A - C, A - B, sẽ tương ứng cho việc **xoay ngược chiều kim đồng hồ của các ổ khóa 4, 6, 8.**
- Do vậy, những ổ khóa 2, 4, 6, 8 có thể được xoay độc lập. Vì thế, ta có thể khiến ổ khóa 2, 4, 6, 8 về cùng với giá trị với ổ khóa 5. Và cuối cùng xoay các ổ khóa 1, 3, 7, 9 về cùng giá trị.



Bài L
Lucky Pair

Số AC (trước đóng băng): 215
AC đầu tiên: 6’ - Meld

Lời giải: Lê Bảo Hiệp
HCMUS-FlamingTomatoes

Bài L - Lucky Pair

Tóm đề:

- Cặp (x, y) được gọi là *lucky pair* nếu tồn tại k nguyên dương thỏa $x^k + y^k$ chia hết cho xy.
- Cho mảng các số nguyên dương. Đếm số cặp phần tử là *lucky pair*.

Bài L - Lucky Pair

Lời giải:

- Gọi S_u là tập thừa số nguyên tố của u.
- $(x + y)^k = x^k + y^k$ + bội của *xy* (định lý nhị thức)
- Điều kiện *lucky pair* tương đương với $(x + y)^k$ chia hết cho xy
  $\rightarrow S_{xy} \subseteq S_{x+y}$
- Dễ thấy: $S_x \subseteq S_{xy} \subseteq S_{x+y} \rightarrow S_x \subseteq S_{x+y} \rightarrow S_x \subseteq S_y$.
- Tương tự: $S_y \subseteq S_x$
- Do đó $S_x = S_y$, hay **x và y có cùng tập thừa số nguyên tố**.

Bài L - Lucky Pair

Lời giải:

- Dùng sàng nguyên tố để tính tích thừa số nguyên tố của mỗi phần tử.
- Nhóm các phần tử theo tích thừa số nguyên tố rồi tính tổng số cặp trong mỗi nhóm.

Bài M
Master Chef

Số AC (trước đóng băng): 72
AC đầu tiên: 57’ - BKDN.Natural

Lời giải: Nguyễn Đình Phúc
Pendle

Bài M - Master Chef

Tóm đề:

- Cho **n ≤ 100**, **n-1** có tối đa **2** ước.
- Tìm và in ra nhiều nhất có thể các tập gồm **n** số nguyên dương phân biệt thỏa mãn:
    - Mỗi số xuất hiện trong tối đa **n** tập.
    - Tất cả các tập đôi một có chung nhau đúng **1** phần tử.

Bài M - Master Chef

Nhận xét:

- Không mất tính tổng quát, giả sử tập thứ nhất chứa các số **(1, 2, 3, ..., n)**.
- Mỗi số từ 1 đến n sẽ xuất hiện trong tối đa **n - 1** tập nữa.

  → Chỉ có tối đa **1 + n (n - 1)** tập có thể tạo ra được

Bài M - Master Chef

Lời giải

- Khi **n=2**, in test đề.

Bài M - Master Chef

Lời giải

- Đầu tiên ta có tập **(1 → n)** và **n - 1** tập mỗi loại bắt đầu từ **(2, 3, ..., n)**

1	2	3	4	...	n
2 2 2 ... (n-1)th 2					
3 3 3 ... (n-1)th 3					
...					

Bài M - Master Chef

Lời giải

- Đầu tiên ta có tập **(1 → n)** và **n - 1** tập mỗi loại bắt đầu từ **(2, 3, ..., n)**
- Điền $\mathbf{(n-1)^2}$ số phân biệt vào các tập chứa số **2**.

1	2	3	4	...	n
2	n+1	n+2	n+3	...	n+n-1
2	2n+1	2n+2	2n+3	...	2n+n-1
2	3n+1	3n+2	3n+3	...	3n+n-1
...	...	...	...	...	...
(n-1)th 2	(n-1)n+1	(n-1)n+2	(n-1)n+3	...	(n-1)n+n-1
3					
3					
3					
...					
(n-1)th 3					
...					

Bài M - Master Chef

Lời giải

- Đầu tiên ta có tập **(1 → n)** và **n - 1** tập mỗi loại bắt đầu từ **(2, 3, ..., n)**
- Điền **$(n-1)^2$** số phân biệt vào các tập chứa số **2**.
- Bắt đầu từ bộ các tập bắt đầu từ **3** trở đi, tập bắt đầu từ **i** sẽ được tạo ra bằng cách:
    - Số thứ k của tập thứ j được thêm vào là: **kn + 1 + ((i-2)*j) % (n-1)**

1	2	3	4	...	n
2	n+1	n+2	n+3	...	n+n-1
2	2n+1	2n+2	2n+3	..	2n+n-1
2	3n+1	3n+2	3n+3	..	3n+n-1
...	...	...	...	...	...
(n-1)th 2	(n-1)n+1	(n-1)n+2	(n-1)n+3	...	(n-1)n+n-1
3	n+1	2n+2	3n+4	...	(n-1)n+n-1
3	n+2	2n+3	3n+3	...	(n-1)n+1
3	n+3	2n+4	3n+5		(n-1)n+2
...	...	...	...	...	...
(n-1)th 3	n+n-1	2n+1	3 n+2	...	(n-1)n+n-2
...					

Bài M - Master Chef

Lời giải

- Đầu tiên ta có tập **(1 → n)** và **n - 1** tập mỗi loại bắt đầu từ **(2, 3, ..., n)**
- Điền **$(n-1)^2$** số phân biệt vào các tập chứa số **2**.
- Bắt đầu từ bộ các tập bắt đầu từ **3** đến **n (% n-1),** tập bắt đầu từ **i** sẽ được tạo ra bằng cách:
  - Số thứ k của tập thứ j được thêm vào là: **kn + 1 + ((i-2)*j) % (n-1)**

1	2	3	4	...	n
2	n+1	n+2	n+3	...	n+n-1
2	2n+1	2n+2	2n+3	..	2n+n-1
2	3n+1	3n+2	3n+3	..	3n+n-1
...	...	...	...	...	...
(n-1)th 2	(n-1)n+1	(n-1)n+2	(n-1)n+3	...	(n-1)n+n-1
4	n+1	2n+3	3n+5	...	(n-1)n+n-2
4	n+2	2n+4	3 n+6	...	(n-1)n+n-1
4	n+3	2n+5	3n+7		(n-1)n+1
...	...	...	...	...	...
(n-1)th 4	n+n-1	2n+2	3 n+4	...	(n-1)n+n-3
...					

Bài M - Master Chef

Chứng minh:

- Các số cùng 1 bộ chung nhau duy nhất 1 số

1	2	3	4	...	n
3	n+1	2n+2	3n+3	...	(n-1)n+n-1
3	n+2	2n+3	3 n+4	...	(n-1)n+1
3	n+3	2n+4	3n+5		(n-1)n+2
...	...	...	...	...	...
(n-1)th 3	n+n-1	2n+1	3 n+2	...	(n-1)n+n-2
4	n+1	2n+3	3n+5	...	(n-1)n+n-2
4	n+2	2n+4	3 n+6	...	(n-1)n+n-1
4	n+3	2n+5	3n+7		(n-1)n+1
...	...	...	...	...	...
(n-1)th 4	n+n-1	2n+2	3 n+4	...	(n-1)n+n-3
...					

Bài M - Master Chef

Chứng minh:

- Các số cùng 1 bộ chung nhau duy nhất 1 số
- Mọi tập đều chung với tập đầu tiên 1 số **< n**

1	2	3	4	...	n
3	n+1	2n+2	3n+3	...	(n-1)n+n-1
3	n+2	2n+3	3n+4	...	(n-1)n+1
3	n+3	2n+4	3n+5		(n-1)n+2
...	...	...	...	...	...
(n-1)th 3	n+n-1	2n+1	3n+2	...	(n-1)n+n-2
4	n+1	2n+3	3n+5	...	(n-1)n+n-2
4	n+2	2n+4	3n+6	...	(n-1)n+n-1
4	n+3	2n+5	3n+7		(n-1)n+1
...	...	...	...	...	...
(n-1)th 4	n+n-1	2n+2	3n+4	...	(n-1)n+n-3
...					

Bài M - Master Chef

Chứng minh:

- Các số cùng 1 bộ chung nhau duy nhất 1 số
- Mọi tập đều chung với tập đầu tiên 1 số **< n**
- Mọi tập thuộc các bộ bắt đầu từ **(3, 4, ... n)**: hàng i và hàng j sẽ chung nhau đúng số thứ **1 + (j-i) % (n-1)**. Không trùng nhiều hơn do n-1 nguyên tố.

1	2	3	4	...	n
3	n+1	2n+2	3n+3	...	(n-1)n+n-1
3	n+2	2n+3	3 n+4	...	(n-1)n+1
3	n+3	$\boxed{2n+4}$	3n+5		(n-1)n+2
...	...	...	...	...	...
(n-1)th 3	n+n-1	2n+1	3 n+2	...	(n-1)n+n-2
4	n+1	2n+3	3n+5	...	(n-1)n+n-2
4	n+2	$\boxed{2n+4}$	3 n+6	...	(n-1)n+n-1
4	n+3	2n+5	3n+7		(n-1)n+1
...	...	...	...	...	...
(n-1)th 4	n+n-1	2n+2	3 n+4	...	(n-1)n+n-3
...					

Bài M - Master Chef

Nhận xét về bài:

- Nên làm tay các test nhỏ để đoán đáp án và tìm pattern.