

## TỔNG QUAN VỀ ĐỀ BÀI

| STT | Tên file bài làm | Tên file dữ liệu | Tên file kết quả | Giới hạn mỗi test | Điểm |
|-----|------------------|------------------|------------------|-------------------|------|
| 1   | ARITHDIG.*       | ARITHDIG.INP     | ARITHDIG.OUT     | 1 giây/1 GB       | 100  |
| 2   | INVERSION.*      | INVERSION.INP    | INVERSION.OUT    | 1 giây/1 GB       | 100  |
| 3   | TREE.*           | TREE.INP         | TREE.OUT         | 1 giây/1 GB       | 100  |
| 4   | REGIONS.*        | REGIONS.INP      | REGIONS.OUT      | 1 giây/1 GB       | 100  |

Dấu \* được thay bằng PAS hay CPP tùy theo ngôn ngữ lập trình được sử dụng là Pascal hay C++

Hãy lập chương trình giải các bài toán sau đây

### Bài 1. Chữ số tuyến tính

Một số nguyên dương được gọi là số có chữ số tuyến tính nếu dãy chữ số của số đó lập thành một cấp số cộng theo modul 10. Chẳng hạn số 15937 là một số có chữ số tuyến tính vì dãy (1,5,9,3,7) là một cấp số cộng công sai 4 trong modul 10.

**Yêu cầu:** Cho hai số nguyên  $l, r$ , đếm số số có chữ số tuyến tính trong phạm vi  $[l; r]$ .

**Dữ liệu:** Vào từ file văn bản ARITHDIG.INP

- ✿ Dòng đầu tiên chứa số nguyên dương  $T$  là số tests,
- ✿ Các dòng tiếp theo, cứ hai dòng liên tiếp chứa dữ liệu cho một test theo định dạng
  - ✿ Dòng 1: số nguyên  $l$
  - ✿ Dòng 2: số nguyên  $r$  ( $1 \leq l \leq r \leq 10^{10^5}$ )

**Kết quả:** Ghi ra file văn bản ARITHDIG.OUT, ứng với mỗi test, ghi ra trên một dòng số số có chữ số tuyến tính trong phạm vi  $[l; r]$ .

**Ví dụ**

| ARITHDIG.INP | ARITHDIG.OUT | Giải thích   |
|--------------|--------------|--|
| 2            | 7            | Test case 1:   |
| 9            | 11           | 7 số là 9, 10, 11, 12, 13, 14, 15                    |
| 15           |              | Test case 2:   |
| 90           |              | 11 số là 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 109 |
| 110          |              |  |

**Chú ý**

20% số điểm ứng với các test có  $1 \leq l \leq r \leq 10^9; r - l \leq 10^5$

20% số điểm ứng với các test có  $l$  và  $r$  cùng số chữ số

20% số điểm ứng với các test có  $l = 1, r = 10^k$

40% số điểm còn lại ứng với các test không có ràng buộc bổ sung

### Bài 2. Nghịch thế

Với  $A = (a_1, a_2, \dots, a_n)$  là một hoán vị của tập hợp các số nguyên dương từ 1 tới  $n: \{1, 2, \dots, n\}$ , ta định nghĩa số lượng nghịch thế của dãy  $A$  là số lượng những cặp chỉ số  $(i, j)$  thỏa mãn:  $i < j$  và  $a_i > a_j$ .

**Yêu cầu:** Cho hai số nguyên dương  $n, m$  hãy tìm hoán vị nhỏ nhất theo thứ tự từ điển của tập  $\{1, 2, \dots, n\}$  sao cho số lượng nghịch thế của hoán vị đó đúng bằng  $m$ .

(Một hoán vị  $A = (a_1, a_2, \dots, a_n)$  được gọi là nhỏ hơn hoán vị  $B = (b_1, b_2, \dots, b_n)$  theo thứ tự từ điển nếu và chỉ nếu tồn tại một chỉ số  $k$  sao cho  $a_k < b_k$  và  $\forall i: 1 \leq i < k$ , ta có  $a_i = b_i$ )

**Dữ liệu:** Vào từ file văn bản INVERSION.INP

- ✿ Dòng đầu tiên chứa số nguyên dương  $T \leq 100$  là số lượng test
- ✿  $T$  dòng tiếp theo, mỗi dòng chứa dữ liệu về 1 test: gồm hai số nguyên  $n, m$  cách nhau bởi dấu cách ( $1 \leq n \leq 10^6; 0 \leq m \leq 10^{18}$ )

**Kết quả:** Ghi ra file văn bản INVERSION.OUT

Ghi ra  $T$  dòng, mỗi dòng ghi một số nguyên duy nhất là mã số của hoán vị tìm được tương ứng với một test. Mã số của một hoán vị  $A = (a_1, a_2, \dots, a_n)$  được tính bằng số dư của phép chia giá trị biểu thức sau cho  $10^9 + 7$ :

$$\sum_{i=1}^n a_i 2^{i-1} = a_1 + a_2 \cdot 2 + a_3 \cdot 2^2 + \dots + a_n \cdot 2^{n-1}$$

Trong trường hợp không tồn tại hoán vị thỏa mãn, trên dòng tương ứng ghi số -1.

**Ví dụ**

| INVERSION.INP | INVERSION.OUT | Giải thích  |
|---------------|---------------|---|
| 2             | 58            | Với $n = 5, m = 9$ , hoán vị cần tìm (4, 5, 3, 2, 1)                                    |
| 5 9           | 673           | $4 + 5 * 2 + 3 * 4 + 2 * 8 + 1 * 16 = 58 \equiv 58 \pmod{10^9 + 7}$                     |
| 7 3           |               | Với $n = 7, m = 3$ , hoán vị cần tìm (1, 2, 3, 4, 7, 6, 5)                              |
|               |               | $1 + 2 * 2 + 3 * 4 + 4 * 8 + 7 * 16 + 6 * 32 + 5 * 64 = 673 \equiv 673 \pmod{10^9 + 7}$ |

**Chú ý**

25% số điểm ứng với các test có  $1 \leq n \leq 10$

25% số điểm ứng với các test có  $1 \leq n \leq 1000$

25% số điểm ứng với các test có  $1 \leq n \leq 10^5$

25% số điểm còn lại ứng với các test không có ràng buộc bổ sung

### Bài 3. Cây biểu thức

Cho một cây nhị phân, trong đó mỗi nút trên cây thuộc một trong ba loại: nút lá, nút max và nút min. Bạn cần gán các số nguyên cho trước vào các nút lá của cây sao cho giá trị của gốc là lớn nhất có thể, trong đó giá trị của các nút được tính như sau:

- ✿ Nút lá có giá trị đúng bằng số được gán vào nút đó
- ✿ Nút max có giá trị bằng giá trị lớn nhất trong các nút con của nó
- ✿ Nút min có giá trị bằng giá trị nhỏ nhất trong các nút con của nó.

**Dữ liệu:** Vào từ file văn bản TREE.INP

- ✿ Dòng 1 chứa hai số nguyên dương  $n, m$ , trong đó  $n$  là số nút trên cây và  $m$  là số nút lá. Các nút được đánh số từ 1 tới  $n$  với nút gốc là đỉnh 1.
- ✿ Dòng 2 chứa  $m$  số nguyên dương  $a_1, a_2, \dots, a_m$  là danh sách các số dùng để gán vào các lá, mỗi phần tử  $a_i$  cần được gán cho duy nhất 1 lá.
- ✿ Dòng thứ  $i$  trong số  $n$  dòng tiếp theo chứa 3 số nguyên  $k_i, l_i, r_i$ , mô tả nút  $i$ :
  - ✿  $k_i \in \{0, 1, 2\}$  cho biết loại nút:  $k_i = 0$  là nút lá,  $k_i = 1$  là nút max,  $k_i = 2$  là nút min
  - ✿  $l_i$  là số hiệu nút con trái của nút  $i$ , quy ước  $l_i = 0$  nếu nút  $i$  không có con trái.
  - ✿  $r_i$  là số hiệu nút con phải của nút  $i$ , quy ước  $r_i = 0$  nếu nút  $i$  không có con phải.

**Kết quả:** Ghi ra file văn bản TREE.OUT một số nguyên duy nhất là giá trị của nút gốc tính theo phương án tìm được

**Ví dụ**

| TREE.INP | TREE.OUT |
|----------|----------|
| 7 4      | 2        |
| 1 2 3 4  |          |
| 2 2 3    |          |
| 1 4 5    |          |
| 2 6 7    |          |
| 0 0 0    |          |
| 0 0 0    |          |
| 0 0 0    |          |
| 0 0 0    |          |

**Chú ý**

30% số điểm ứng với các test thỏa mãn:  $n \leq 27$ , đồng thời mọi nút min và nút max có đúng 2 con

70% số điểm còn lại ứng với các test thỏa mãn:  $n \leq 1000$

## Bài 4. Sơ đồ giao thông

Những năm gần đây, ngày càng có nhiều khách du lịch đến thăm tỉnh Bắc Ninh. Có những du khách đến để ngắm vẻ đẹp thiên nhiên trời phú cho Bắc Ninh và cũng có không ít người lại đến để tìm hiểu hệ thống giao thông rất đặc biệt của Bắc Ninh.

Bắc Ninh có  $n$  thành phố, được đánh số từ 1 đến  $n$ . Hệ thống đường giao thông của tỉnh gồm  $m$  đoạn đường hai chiều. Mỗi đoạn đường nối một cặp gồm hai thành phố khác nhau, không có hai đoạn đường nào nối cùng một cặp thành phố. Các đoạn đường này không nhất thiết đảm bảo việc đi lại giữa hai thành phố bất kỳ, vì thế Bắc Ninh được chia ra thành các vùng. Mỗi vùng gồm một nhóm các thành phố sao cho từ một thành phố bất kỳ trong vùng có thể đi sang mọi thành phố còn lại trong vùng (có thể phải qua một số thành phố trung gian), nhưng không thể đến bất cứ thành phố nào nằm ngoài vùng. Vùng có thể chỉ gồm duy nhất một thành phố.

Chính quyền Bắc Ninh lập dự án xây dựng bổ sung thêm  $k$  đoạn đường mới, mỗi đoạn nối một cặp thành phố khác nhau. Thêm vào đó, giữa hai thành phố bất kỳ vẫn phải đảm bảo không được có nhiều hơn 1 đoạn đường nối trực tiếp giữa chúng. Sau khi xây dựng xong các đoạn đường mới, việc phân các thành phố vào các vùng sẽ thay đổi và số vùng cũng thay đổi.

Ví dụ: Với  $n = 6$ , ban đầu có hai đoạn đường nối hai thành phố 1 và 2 và nối hai thành phố 3 và 4 (thể hiện bởi đường liền nét trong hình minh họa). Khi đó có 4 vùng: vùng thứ nhất gồm hai thành phố 1 và 2, vùng thứ hai gồm hai thành phố 3 và 4, vùng thứ ba gồm thành phố 5, vùng thứ tư gồm thành phố 6. Nếu xây dựng thêm  $k = 3$  đoạn đường giữa hai thành phố 1 và 4, giữa 2 và 4, giữa 2 và 3 (các đoạn đường này được thể hiện bởi đường nối đứt nét trong hình minh họa), số lượng vùng giảm đi 1 (hai vùng thứ nhất và thứ hai được nối lại thành một vùng).

Để phê duyệt dự án cải tổ hệ thống giao thông, còn phải chỉ ra cụ thể cần xây dựng  $k$  đoạn đường nào. Để làm điều này, trước hết chính quyền Bắc Ninh muốn xác định số vùng nhỏ nhất và số vùng lớn nhất hình thành sau khi xây dựng thêm  $k$  đoạn đường thỏa mãn những điều kiện đã nêu.

**Yêu cầu:** Cho biết sơ đồ giao thông hiện tại của Bắc Ninh và số nguyên dương  $k$ , hãy xác định xem nếu xây dựng bổ sung thêm  $k$  đoạn đường thì số vùng nhỏ nhất và số vùng lớn nhất có thể là bao nhiêu.

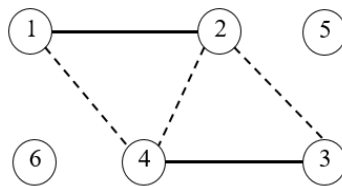
**Dữ liệu:** Vào từ file văn bản REGIONS.INP:

- ✿ Dòng đầu tiên chứa ba số nguyên được ghi cách nhau bởi dấu cách  $n, m$  và  $k$  ( $2 \leq n \leq 10^5, 0 \leq m \leq 10^5; 1 \leq k \leq \min\{10^9, \frac{n(n-1)}{2} - m\}$ ) theo thứ tự là số lượng thành phố, số lượng đoạn đường trong hệ thống giao thông hiện tại và số lượng đoạn đường sẽ xây dựng thêm theo dự án cải tổ.
- ✿ Mỗi dòng trong số  $m$  dòng tiếp theo chứa hai số nguyên  $u$  và  $v$  ( $1 \leq u, v \leq n$ ) được ghi cách nhau bởi dấu cách cho biết có đoạn đường hai chiều nối hai thành phố  $u$  và  $v$ .

**Kết quả:** Ghi ra file văn bản REGIONS.OUT hai số nguyên cách nhau bởi dấu cách theo thứ tự là số lượng nhỏ nhất và số lượng lớn nhất vùng có thể hình thành khi hoàn thành dự án.

**Ví dụ**

| REGIONS.INP         | REGIONS.OUT |
|---------------------|-------------|
| 6 2 3<br>1 2<br>3 4 | 1 3         |
| 3 1 1<br>2 1        | 1 1         |



∞ HẾT ∞