

# CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT – CONTEST 5

## HÀNG ĐỘI – SẮP XẾP

### A. CẤU TRÚC DỮ LIỆU HÀNG ĐỘI 1

Ban đầu cho một queue rỗng. Bạn cần thực hiện các truy vấn sau:

1. Trả về kích thước của queue
2. Kiểm tra xem queue có rỗng không, nếu có in ra “YES”, nếu không in ra “NO”.
3. Cho một số nguyên và đẩy số nguyên này vào cuối queue.
4. Loại bỏ phần tử ở đầu queue nếu queue không rỗng, nếu rỗng không cần thực hiện.
5. Trả về phần tử ở đầu queue, nếu queue rỗng in ra -1.
6. Trả về phần tử ở cuối queue, nếu queue rỗng in ra -1.

#### Dữ liệu vào

Dòng đầu tiên chứa số nguyên T là số bộ dữ liệu, mỗi bộ dữ liệu theo dạng sau.

Dòng đầu tiên chứa số nguyên n - lượng truy vấn ( $1 \leq n \leq 1000$ )

N dòng tiếp theo, mỗi dòng sẽ ghi loại truy vấn như trên, với truy vấn loại 3 sẽ có thêm một số nguyên, không quá  $10^6$ .

**Kết quả:** In ra kết quả của các truy vấn..

#### Ví dụ:

Input	Output
1	1
1 4	3
3 1	5
3 2	2
3 3	
5	
6	
4	
4	
4	
4	
4	
3 5	
3 6	
5	
1	

### B. CẤU TRÚC DỮ LIỆU HÀNG ĐỘI 2

Yêu cầu bạn xây dựng một queue với các truy vấn sau đây:

“PUSH x”: Thêm phần tử x vào cuối của queue ( $0 \leq x \leq 1000$ ).

“PRINTFRONT”: In ra phần tử đầu tiên của queue. Nếu queue rỗng, in ra “NONE”.

“POP”: Xóa phần tử ở đầu của queue. Nếu queue rỗng, không làm gì cả.

**Dữ liệu vào:**

Dòng đầu tiên là số lượng truy vấn  $Q$  ( $Q \leq 100000$ ).

Mỗi truy vấn có dạng như trên.

**Kết quả:**

Với mỗi truy vấn “PRINT”, hãy in ra phần tử đầu tiên của queue. Nếu queue rỗng, in ra “NONE”.

**Ví dụ:**

Input	Output
9	2
PUSH 1	2
PUSH 2	NONE
POP	
PRINTFRONT	
PUSH 3	
PRINTFRONT	
POP	
POP	
PRINTFRONT	

### C. HÀNG ĐỢI HAI ĐẦU (DEQUEUE)

Yêu cầu bạn xây dựng một hàng đợi hai đầu với các truy vấn sau đây:

“PUSHFRONT x”: Thêm phần tử  $x$  vào đầu của dequeue ( $0 \leq x \leq 1000$ ).

“PRINTFRONT”: In ra phần tử đầu tiên của dequeue. Nếu dequeue rỗng, in ra “NONE”.

“POPFRONT”: Xóa phần tử đầu của dequeue. Nếu dequeue rỗng, không làm gì cả.

“PUSHBACK x”: Thêm phần tử  $x$  vào cuối của dequeue ( $0 \leq x \leq 1000$ ).

“PRINTBACK”: In ra phần tử cuối của dequeue. Nếu dequeue rỗng, in ra “NONE”.

“POPBACK”: Xóa phần tử cuối của dequeue. Nếu dequeue rỗng, không làm gì cả.

**Dữ liệu vào:**

Dòng đầu tiên là số lượng truy vấn  $Q$  ( $Q \leq 100000$ ).

Mỗi truy vấn có dạng như trên.

**Kết quả:**

Với mỗi truy vấn “PRINTFRONT” và “PRINTBACK”, hãy in ra kết quả trên một dòng.

**Ví dụ:**

Input	Output
10	2
PUSHBACK 1	1
PUSHFRONT 2	3
PUSHBACK 3	NONE
PRINTFRONT	

POPFONT	
PRINTFONT	
POPFONT	
PRINTBACK	
POPFONT	
PRINTBACK	

## D. ĐƯỜNG NGUYÊN TỐ

Cho hai số nguyên tố khác nhau có bốn chữ số. Người ta cho rằng hoàn toàn có thể biến đổi từ số này thành số kia sau một số bước theo quy tắc: Tại mỗi bước ta chỉ thay đổi một chữ số trong số trước đó sao cho số tạo được trong mỗi bước đều là một số nguyên tố có bốn chữ số. Một cách biến đổi như vậy gọi là một “đường nguyên tố”.

Bài toán đặt ra là với một cặp số nguyên tố đầu vào, hãy tính ra số bước của đường nguyên tố ngắn nhất. Giả sử đầu vào là hai số 1033 và 8179 thì đường nguyên tố ngắn nhất sẽ có độ dài là 6 với các bước chuyển là:

1033  
1733  
3733  
3739  
3779  
8779  
8179

**Dữ liệu vào:** Dòng đầu tiên ghi số bộ test, không lớn hơn 100. Mỗi bộ test viết trên một dòng bao gồm hai số nguyên tố có 4 chữ số..

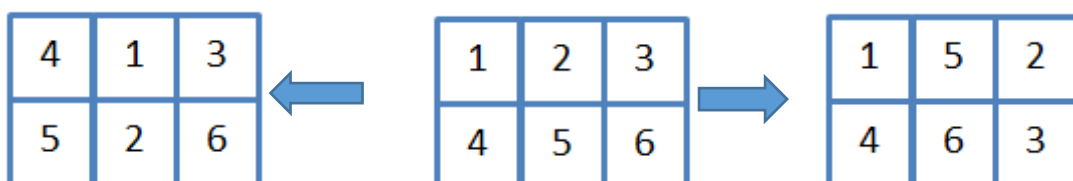
**Kết quả:** Với mỗi bộ test, in ra màn hình trên một dòng số bước của đường nguyên tố ngắn nhất.

**Ví dụ:**

Input	Output
3	6
1033 8179	7
1373 8017	0
1033 1033	

## E. QUAY HÌNH VUÔNG

Có một chiếc bảng hình chữ nhật với 6 miếng ghép, trên mỗi miếng ghép được điền một số nguyên trong khoảng từ 1 đến 6. Tại mỗi bước, chọn một hình vuông (bên trái hoặc bên phải), rồi quay theo chiều kim đồng hồ.



**Yêu cầu:** Cho một trạng thái của bảng, hãy tính số phép biến đổi ít nhất để đưa bảng đến trạng thái đích.

**Dữ liệu vào:**

Dòng đầu tiên chứa 6 số là trạng thái bảng ban đầu (thứ tự từ trái qua phải, dòng 1 tới dòng 2).

Dòng thứ hai chứa 6 số là trạng thái bảng đích (thứ tự từ trái qua phải, dòng 1 tới dòng 2).

**Kết quả:**

In ra một số nguyên là đáp số của bài toán.

**Ví dụ:**

Input	Output
1 2 3 4 5 6 4 1 2 6 5 3	2

## F. DI CHUYỂN

Cho một bảng kích thước  $N \times N$ , trong đó có các ô trống '.' và vật cản 'X'. Các hàng và các cột được đánh số từ 0.

Mỗi bước di chuyển, bạn có thể đi từ ô  $(x, y)$  tới ô  $(u, v)$  nếu như 2 ô này nằm trên cùng một hàng hoặc một cột, và không có vật cản nào ở giữa.

Cho điểm xuất phát và điểm đích. Bạn hãy tính số bước di chuyển ít nhất?

**Dữ liệu vào:**

Dòng đầu tiên là số nguyên dương  $N$  ( $1 \leq N \leq 100$ ).

$N$  dòng tiếp theo, mỗi dòng gồm  $N$  kí tự mô tả bảng.

Cuối cùng là 4 số nguyên  $a, b, c, d$  với  $(a, b)$  là tọa độ điểm xuất phát,  $(c, d)$  là tọa độ đích. Dữ liệu đảm bảo hai vị trí này không phải là ô cấm.

**Kết quả:**

In ra một số nguyên là đáp số của bài toán.

**Ví dụ:**

Input	Output
3 .X. .X. ... 0 0 0 2	3

## G. GIEO MÀM

Trên một giá có kích thước  $R \times C$  ( $R$  hàng,  $C$  cột), một số hạt mầm đã được tra vào các ô. Một số hạt mầm được bón thêm chất dinh dưỡng, nên đã nảy mầm sớm thành cây non.

Mỗi ngày, các cây non sẽ lan truyền chất dinh dưỡng của nó cho các mầm ở ô xung quanh (trái, trên, phải, dưới), làm cho các hạt mầm này phát triển thành cây non. Tuy nhiên, có thể có một số hạt mầm được gieo ở vị trí lẻ loi, do không nhận được chất dinh dưỡng nên không thể nảy mầm.

Các bạn hãy xác định xem cần ít nhất bao nhiêu ngày để tất cả các hạt đều mầm?

#### Dữ liệu vào:

Dòng đầu tiên gồm 2 số nguyên  $R$  và  $C$  ( $1 \leq R, C \leq 500$ ).

$R$  dòng tiếp theo, mỗi dòng gồm  $C$  số nguyên  $A[i][j]$ .

$A[i][j] = 0$ , ô  $(i, j)$  là ô trống.

$A[i][j] = 1$ , ô  $(i, j)$  là hạt chưa nảy mầm.

$A[i][j] = 2$ , ô  $(i, j)$  là cây non.

#### Kết quả:

In ra thời gian ngắn nhất để tất cả các hạt đều nảy mầm. Nếu có hạt nào chưa nảy mầm, in ra -1.

#### Ví dụ:

Test 1	Test 2
Input: 3 5 2 1 0 2 1 1 0 1 2 1 1 0 0 2 1  Output: 2	Input: 3 5 2 1 0 2 1 0 0 1 2 1 1 0 0 2 1  Output: -1

## H. DI CHUYỂN TRONG KHÔNG GIAN

Cho một hình hộp chữ nhật có kích thước  $A \times B \times C$ , trong đó  $A$  là chiều cao,  $B$  là chiều rộng và  $C$  là chiều dài. Mỗi ô có thể là một ô trống '.' hoặc vật cản '#'.

Mỗi bước, bạn được phép di chuyển sang một ô kề bên cạnh (không được đi chéo). Nhiệm vụ của bạn là tìm đường đi ngắn nhất bắt đầu 'S' tới vị trí kết thúc 'E'.

#### Dữ liệu vào:

Dòng đầu tiên là số lượng bộ test  $T$  ( $1 \leq T \leq 50$ ).

Mỗi test bắt đầu bởi 3 số nguyên  $A, B, C$  ( $A, B, C \leq 30$ ).

Tiếp theo là  $A$  khối, mỗi khối gồm  $B \times C$  kí tự mô tả một lát cắt của hình hộp chữ nhật. Giữa 2 khối có một dấu xuống dòng.

#### Kết quả:

In ra một số nguyên là đường đi ngắn nhất từ S tới E. Nếu không di chuyển được, in ra -1.

#### Ví dụ:

Input	Output
2 3 4 5 S . . . . . # # # . . # # . . # # # . #  # # # # #	11 -1

##### ##.## ##...  ##### ##### #.### #####E  1 3 3 S## #E# ###	
--	--

## I. TÌM BỘỊ SỐ LỚN NHẤT CỦA 3

Cho dãy số  $A[]$  có  $N$  phần tử là các chữ số từ 0 đến 9. Nhiệm vụ của bạn là hãy chọn lấy một tổ hợp các phần tử và sắp xếp chúng sao cho thu được số lớn nhất chia hết cho 3.

Nếu không tìm được số nào, in ra -1.

### Dữ liệu vào:

Dòng đầu tiên là số lượng bộ test  $T$  ( $1 \leq T \leq 50$ ).

Mỗi test bắt đầu bởi số nguyên  $N$  ( $1 \leq N \leq 100\,000$ ).

Dòng tiếp theo gồm  $N$  số nguyên  $A[i]$  ( $0 \leq A[i] \leq 9$ ).

### Kết quả:

Với mỗi test, in ra đáp án tìm được trên một dòng.

### Ví dụ:

Input	Output
3	981
3	8760
8 1 9	-1
5	
8 1 7 6 0	
2	
5 2	

## J. SỐ BDN

Ta gọi số nguyên dương  $K$  là một số BDN nếu các chữ số trong  $K$  chỉ bao gồm các 0 hoặc 1 có nghĩa. Ví dụ số  $K = 1, 10, 101$ . Cho số tự nhiên  $N$  ( $N < 2^{63}$ ). Hãy cho biết có bao nhiêu số BDN không lớn hơn  $N$ . Ví dụ  $N=100$  ta có 4 số BDN bao gồm các số: 1, 10, 11, 100.

**Dữ liệu vào:** Dòng đầu tiên ghi lại số tự nhiên  $T$  ( $T \leq 100$ ) là số lượng bộ test.

$T$  dòng kế tiếp mỗi dòng là một số tự nhiên  $N$  ( $1 \leq N \leq 2^{63}$ ).

**Kết quả:** Với mỗi số  $N$  in ra kết quả trên một dòng là số các số BDN không lớn hơn  $N$ .

Input	Output
3	2
10	4
100	7
200	

## K. HEXGAME

HEXGAME là một trò chơi xếp hình gồm 10 miếng ghép hình lục giác đều, trên mỗi miếng ghép được điền một số nguyên, có 8 miếng ghép được điền số từ 1 đến 8 và có hai miếng ghép được điền số 0. Các miếng ghép liên kết với nhau tạo thành lưới tổ ong. Ban đầu các miếng ghép ở vị trí như hình vẽ. Tại mỗi bước, chọn một miếng ghép có đúng 6 miếng ghép kề cạnh làm tâm, rồi xoay một nấc 6 miếng ghép kề cạnh đó theo chiều kim đồng hồ. Như vậy chỉ có hai cách chọn tâm, đó là chọn tâm bên trái và chọn tâm bên phải.



**Yêu cầu:** Cho một trạng thái của trò chơi (nhận được sau một dãy biến đổi từ trạng thái ban đầu), hãy tính số phép biến đổi ít nhất để đưa về trạng thái ban đầu.

**Dữ liệu vào:**

Dòng đầu tiên chứa 3 số ở 3 miếng ghép dòng thứ nhất (thứ tự từ trái qua phải).

Dòng đầu tiên chứa 4 số ở 4 miếng ghép dòng thứ hai (thứ tự từ trái qua phải).

Dòng đầu tiên chứa 3 số ở 3 miếng ghép dòng thứ ba (thứ tự từ trái qua phải).

**Kết quả:**

In ra một số nguyên là số phép biến đổi ít nhất để đưa được về trạng thái ban đầu.

**Ví dụ:**

Input	Output
1 0 2 8 6 0 3 7 5 4	5

## L. SẮP XẾP ĐỔI CHỖ TRỰC TIẾP

Hãy thực hiện thuật toán sắp xếp đổi chỗ trực tiếp trên dãy N số nguyên. Ghi ra các bước thực hiện thuật toán. **Dữ liệu vào:** Dòng 1 ghi số N (không quá 100). Dòng 2 ghi N số nguyên dương (không quá 100). **Kết quả:** Ghi ra màn hình từng bước thực hiện thuật toán. Mỗi bước trên một dòng, các số trong dãy cách nhau đúng một khoảng trống.

**Ví dụ:**

Input	Output
4 5 7 3 2	Buoc 1: 2 7 5 3 Buoc 2: 2 3 7 5 Buoc 3: 2 3 5 7

## M. SẮP XẾP CHỌN

Hãy thực hiện thuật toán sắp xếp chọn trên dãy N số nguyên. Ghi ra các bước thực hiện thuật toán.

**Dữ liệu vào:** Dòng 1 ghi số N (không quá 100). Dòng 2 ghi N số nguyên dương (không quá 100).

**Kết quả:** Ghi ra màn hình từng bước thực hiện thuật toán. Mỗi bước trên một dòng, các số trong dãy cách nhau đúng một khoảng trống.

**Ví dụ:**

Input	Output
4	Buoc 1: 2 7 3 5
5 7 3 2	Buoc 2: 2 3 7 5
	Buoc 3: 2 3 5 7

## N. SẮP XẾP CHÈN

Hãy thực hiện thuật toán sắp xếp chèn trên dãy N số nguyên. Ghi ra các bước thực hiện thuật toán.

**Dữ liệu vào:** Dòng 1 ghi số N (không quá 100). Dòng 2 ghi N số nguyên dương (không quá 100).

**Kết quả:** Ghi ra màn hình từng bước thực hiện thuật toán. Mỗi bước trên một dòng, các số trong dãy cách nhau đúng một khoảng trống.

**Ví dụ:**

Input	Output
4	Buoc 0: 5
5 7 3 2	Buoc 1: 5 7
	Buoc 2: 3 5 7
	Buoc 3: 2 3 5 7

## O. SẮP XẾP NỔI BỌT

Hãy thực hiện thuật toán sắp xếp nổi bọt trên dãy N số nguyên. Ghi ra các bước thực hiện thuật toán.

**Dữ liệu vào:** Dòng 1 ghi số N (không quá 100). Dòng 2 ghi N số nguyên dương (không quá 100).

**Kết quả:** Ghi ra màn hình từng bước thực hiện thuật toán. Mỗi bước trên một dòng, các số trong dãy cách nhau đúng một khoảng trống.

**Ví dụ:**

Input	Output
4	Buoc 1: 3 2 5 7
5 3 2 7	Buoc 2: 2 3 5 7

## P. SẮP XẾP NHANH

Cho dãy số A[] gồm có N phần tử. Nhiệm vụ của bạn là sắp xếp dãy số theo thứ tự tăng dần.



**Dữ liệu vào:** Dòng đầu tiên là số lượng bộ test  $T$  ( $T \leq 10$ ).

Mỗi test bắt đầu bằng số nguyên  $N$ , ( $N \leq 100\,000$ )

Dòng tiếp theo gồm  $N$  số nguyên  $A[i]$  ( $0 \leq A[i] \leq 10^6$ ).

**Kết quả:**

In ra các phần tử của dãy số sau khi được sắp xếp.

**Ví dụ:**

Input	Output
1 5 2 4 1 3 5	1 2 3 4 5

## Q. SẮP XẾP KHÔNG NHANH

Cho dãy số  $A[]$  gồm có  $N$  phần tử. Nhiệm vụ của bạn là sắp xếp dãy số theo thứ tự tăng dần.

Bộ test được xây dựng để bạn không thể AC nếu sử dụng các phiên bản của sắp xếp nhanh (Quick Sort).

**Dữ liệu vào:**

Mỗi test bắt đầu bằng số nguyên  $N$ , ( $N \leq 100\,000$ )

Dòng tiếp theo gồm  $N$  số nguyên  $A[i]$  ( $0 \leq A[i] \leq 10^{18}$ ).

**Kết quả:**

In ra các phần tử của dãy số sau khi được sắp xếp.

**Ví dụ:**

Input	Output
5 2 4 1 3 5	1 2 3 4 5

## R. SẮP XẾP LẠI DẠY CON

Cho dãy số  $A[]$  gồm có  $N$  phần tử. Nhiệm vụ của bạn là hãy tìm một dãy con liên tiếp dài nhất, sao cho sau khi sắp xếp lại dãy con này, dãy số  $A[]$  sẽ là một dãy số được sắp xếp tăng dần.

**Dữ liệu vào:**

Dòng đầu tiên là số lượng bộ test  $T$  ( $T \leq 10$ ).

Mỗi test bắt đầu bằng số nguyên  $N$  ( $N \leq 100\,000$ ).

Dòng tiếp theo gồm  $N$  số nguyên  $A[i]$  ( $0 \leq A[i] \leq 10^6$ ).

**Kết quả:**

Với mỗi test in ra độ dài dãy con tìm được trên một dòng. Nếu dãy đã cho đã được sắp xếp, in ra "YES".

**Ví dụ:**

Input	Output
3	4
9	2
0 1 15 25 6 7 30 40 50	YES
5	
1 2 4 3 5	
4	
1 2 3 4	

Giải thích test 1: Cần sắp xếp lại dãy con 15, 25, 6, 7.

## S. BRT

Thành phố X có N thị trấn trên trục đường chính. Tọa độ của các thị trấn lần lượt là  $a[1], a[2], \dots, a[N]$ , các tọa độ này là phân biệt, không có 2 tọa độ nào trùng nhau.

Chính quyền thành phố muốn xây dựng một tuyến buýt nhanh BRT để kết nối 2 thị trấn gần nhau nhất với nhau.

Bạn hãy tính thử xem chiều dài của tuyến buýt này bằng bao nhiêu? Và có bao nhiêu cặp thị trấn có tiềm năng giống nhau để xây dựng tuyến BRT này.

### Dữ liệu vào:

Dòng đầu tiên là số lượng bộ test T ( $T \leq 10$ ).

Mỗi test bắt đầu bằng số nguyên N ( $N \leq 100\,000$ ).

Dòng tiếp theo gồm N số nguyên  $A[i]$  ( $-10^9 \leq A[i] \leq 10^9$ ).

### Kết quả:

Với mỗi test in ra 2 số nguyên C và D, lần lượt là khoảng cách ngắn nhất giữa 2 thị trấn, và số lượng cặp thị trấn có cùng khoảng cách ngắn nhất này.

### Ví dụ:

Input	Output
2	2 1
4	2 2
6 -3 0 4	
3	
-2 0 2	

Giải thích test 2: Cặp thị trấn (1, 2) và (2, 3) có cùng khoảng cách.

## T. KHUYẾN MẠI

Một cửa hàng thời trang đang thực hiện chương trình khuyến mại giảm giá. Ban đầu, giá của sản phẩm  $i$  là  $a[i]$ , khi đến tuần giảm giá, giá của chúng giảm xuống còn  $b[i]$ . Tuy nhiên, nhằm đánh lừa khách hàng, mỗi số sản phẩm giá tăng lên chứ không hề giảm xuống.

Nhận biết được quy luật này, Tí mặc dù cần phải mua tổng cộng  $N$  sản phẩm, nhưng cậu quyết định mua  $K$  sản phẩm trước đợt khuyến mại, và  $N-K$  sản phẩm trong đợt khuyến mại.

Giả sử rằng Tí chọn tối ưu được  $K$  sản phẩm ban đầu, các bạn hãy tính xem số tiền ít nhất Tí cần bỏ ra để mua đủ  $N$  sản phẩm là bao nhiêu?

### Dữ liệu vào:

Mỗi test bắt đầu bằng số nguyên  $N$  và  $K$  ( $1 \leq N, K \leq 100\,000$ ).

Dòng thứ hai gồm  $N$  số nguyên  $a[i]$ , giá sản phẩm thứ  $i$  mà trước đợt giảm giá.

Dòng cuối gồm  $N$  số nguyên  $b[i]$ , là giá của sản phẩm sau khi giảm giá.

( $1 \leq a[i], b[i] \leq 10^4$ ).

### Kết quả:

In ra một số nguyên là đáp án của bài toán.

### Ví dụ:

Test 1	Test 2
Input: 3 1 5 4 6 3 1 5  Input: 10	Output: 5 4 3 4 7 10 3 4 5 5 12 5  Output: 25

Giải thích test 1: Tí mua sản phẩm 3 trước khi giảm giá, và sản phẩm 1, 2 trong thời gian khuyến mại.

Giải thích test 2: Tí mua sản phẩm 1, 2, 4, 5 trước, tới đợt khuyến mại thì mua sản phẩm 3.