

SUMMER CONTEST 2020

Lưu ý: Các thí sinh có thể nộp bằng ngôn ngữ Pascal, C++ hoặc Java tùy theo ngôn ngữ nào thí sinh thấy thuận tiện. Tên các bài là Tenbai.*, trong đó * tương ứng với pas, cpp hay java. Tenbai là tên của bài sẽ được quy định trong cụ thể từng bài.

Bài 4 (SC4.*):

Bạn được cho 3 số nguyên dương x , N và M . Hãy tính giá trị $P = (1 + x + x^2 + x^3 + \dots + x^N) \bmod M$.

- **Dữ liệu đầu vào (SC4.INP):**
 - Đọc từ file SC4.INP
 - Dòng đầu tiên gồm một số nguyên dương T ($1 \leq T \leq 40$) là số lượng bộ dữ liệu của đề bài.
 - T bộ dữ liệu tiếp theo, mỗi bộ dữ liệu được tổ chức gồm một dòng duy nhất, chứa 3 số nguyên dương x , N , M tương ứng như phát biểu của đề bài.
- **Dữ liệu đầu ra (SC4.OUT):**
 - Ghi ra file SC4.OUT
 - Với bộ dữ liệu thứ i , ta in ra 1 dòng duy nhất là một số nguyên không âm tương ứng là kết quả của giá trị P tương ứng với bộ test hiện tại.
- **Ví dụ:**

SC4.INP	SC4.OUT
4	13
2 5 50	0
3 1 4	14
5 3 142	744623544
23 57 1000000000	

- **Giải thích test ví dụ:**
 - Với test ví dụ 1: $P = 1 + 2 + 2^2 + 2^3 + 2^4 + 2^5 = 63$. $63 \bmod 50 = 13$.
 - Với test ví dụ 2: $P = 1 + 3 = 4 \bmod 4 = 0$.
 - Với test ví dụ 4: Để giúp bạn đọc có thể kiểm tra lại thuật toán của mình đã chính xác hay chưa.

- **Bài toán có 2 Dataset:**
 - Small Dataset: $1 \leq N \leq 100000$.
 - Large Dataset: $1 \leq N \leq 10^9$.
 - Trong mọi Dataset, ta luôn có $1 \leq x, M \leq 1000000000$.
- **Giới hạn thời gian và bộ nhớ:**
 - 3s / Dataset.
 - 512Mb / Dataset.

Bài 5 (SC5.*):

Bạn được cho một số nguyên dương N. Bạn có 2 thao tác và mỗi lần chỉ có thể sử dụng 1 trong 2 thao tác.

- Thao tác 1: Bạn có thể tăng giá trị của số nguyên hiện tại lên 1 đơn vị. Chi phí để sử dụng thao tác này là A.
- Thao tác 2: Bạn có thể tăng giá trị của số nguyên hiện tại lên 2 lần. Chi phí để sử dụng thao tác này là B.

Bạn hãy sử dụng các thao tác này để chuyển đổi từ số 0 thành số N với chi phí ít nhất.

- **Dữ liệu đầu vào (SC5.INP):**
 - Đọc từ file SC5.INP.
 - Dòng đầu tiên gồm một số nguyên dương T ($1 \leq T \leq 40$) là số lượng bộ dữ liệu cần xử lý
 - T bộ dữ liệu tiếp theo, mỗi bộ dữ liệu chỉ gồm một dòng gồm 3 số nguyên dương N, A, B.
- **Dữ liệu đầu ra (SC5.OUT):**
 - Ghi vào file SC5.OUT.
 - Gồm T dòng, dòng thứ i là kết quả của bộ dữ liệu thứ i, là chi phí ít nhất để có thể biến đổi số 0 thành số N dựa vào các thao tác trên.
- **Ví dụ:**

SC5.INP	SC5.OUT
4	4
2 2 3	6
3 2 3	9
5 2 3	9
6 2 3	

- **Giải thích test ví dụ:**

- Ví dụ 1: Tăng số 0 lên 1 đơn vị, chi phí 2. Tăng 1 lên 1 đơn vị, chi phí 2. Tổng chi phí là 4.
- Ví dụ 2: Ta sử dụng 3 thao tác 1 để tăng số 0 lên 3. Tổng chi phí là $2 * 3 = 6$.
- Ví dụ 3: Ta sử dụng 2 thao tác 1 để tăng 0 lên 2. Sau đó sử dụng thao tác 2 tăng 2 thành 4. Sau đó sử dụng thao tác 1 tăng 4 lên 5. Tổng chi phí là $2 * 2 + 3 + 2 = 9$.
- Ví dụ 4: Ta sử dụng 3 thao tác 1 để tăng 0 lên 3. Sau đó sử dụng thao tác 2 để gấp đôi 3 thành 6. Tổng chi phí là $2 * 3 + 3 = 9$.

- **Bài toán sẽ được chia thành 2 Dataset:**

- Small Dataset: $1 \leq N \leq 500$.
- Large Dataset: $1 \leq N \leq 100000$.
- Trong mọi Dataset, ta luôn có $1 \leq a, b \leq 1000000$

- **Giới hạn dữ liệu:**

- 5s / Dataset.
- 1024Mb / Dataset.

Bài 6 (SC6.*):

Bạn được cho N đỉnh trên một đồ thị và giữa các đỉnh chưa có cạnh nối nào. N đỉnh được đánh số thứ tự từ đỉnh 1 đến đỉnh i. Đỉnh i có màu là c_i , trong đó nếu $c_i = 0$ thì đỉnh i có màu trắng, và $c_i = 1$ thì đỉnh i có màu đen. Ta chỉ được nối thêm cạnh giữa đỉnh có chỉ số nhỏ hơn đến đỉnh có chỉ số lớn hơn. Một “đường đi đen – trắng” là một đường đi xuất phát từ một đỉnh u đến một đỉnh v sao cho chỉ sử dụng cạnh nối trên đồ thị, đồng thời giữa 2 đỉnh liên kề trên đường đi phải khác màu nhau. Bạn được cho thêm một số nguyên p. Bạn hãy đếm có bao nhiêu cách nối thêm cạnh giữa các đỉnh sao cho số lượng “đường đi đen – trắng” có phần dư cho 2 là p.

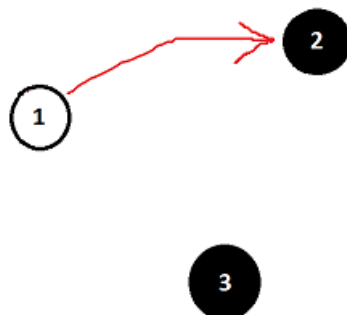
- **Dữ liệu đầu vào (SC6.INP):**

- Dòng đầu tiên gồm một số nguyên dương T ($1 \leq T \leq 40$), là số lượng bộ dữ liệu.
- T bộ dữ liệu tiếp theo, mỗi bộ dữ liệu được tổ chức dưới dạng như sau:
 - Dòng đầu tiên gồm 2 số nguyên N và p, tương ứng là số lượng đỉnh trên đồ thị và p như đã phát biểu trong đề bài.

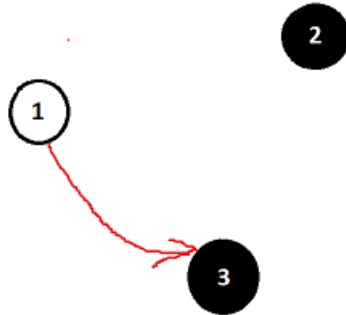
- Dòng thứ 2 gồm N số nguyên, số nguyên thứ i tương ứng là c_i là màu của đỉnh thứ i.
- **Dữ liệu đầu ra (SC6.OUT):**
 - Ta in ra đáp án với T bộ dữ liệu tương ứng, mỗi bộ dữ liệu là một số nguyên, thể hiện đáp án là số lượng cách nối sao cho thỏa mãn đề bài. Vì số cách nối có thể rất lớn nên chỉ đưa ra phần dư của kết quả cho $10^9 + 7$.
- **Ví dụ:**

SC6.INP	SC6.OUT
5	4
3 0	0
0 1 1	2
3 0	32
1 1 1	917504
3 1	
0 1 0	
4 0	
0 1 1 1	
7 1	
1 0 1 1 0 0 1	

- **Giải thích test ví dụ:**
 - Với test ví dụ 1: Ta có 4 cách như sau

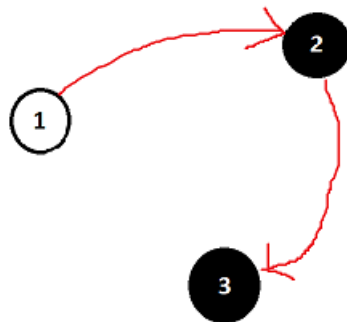


- Cách 1:
 - Có tổng cộng 4 đường đi đen trắng, đó là $\{1\}$, $\{2\}$, $\{3\}$ và $\{1, 2\}$



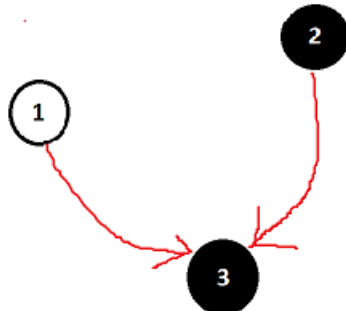
○ Cách 2:

Có tổng cộng 4 đường đi đen trắng, là $\{1\}$, $\{2\}$, $\{3\}$, $\{1, 3\}$



○ Cách 3:

Có tổng cộng 4 đường đi đen trắng, là $\{1\}$, $\{2\}$, $\{3\}$, $\{1, 2\}$



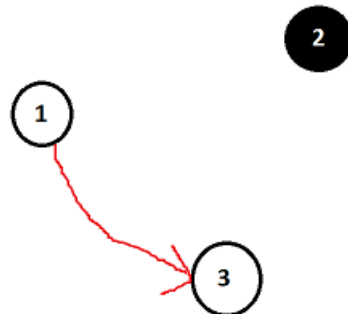
○ Cách 4:

Có tổng cộng 4 đường đi đen trắng, là $\{1\}$, $\{2\}$, $\{3\}$, $\{1, 3\}$

- Với test ví dụ số 2: Cả 3 đỉnh đều có màu đen, do đó dù có nối cạnh nào đi chăng nữa, ta vẫn chỉ có 3 đường đi đen trắng là $\{1\}$, $\{2\}$, $\{3\}$. Trong khi ta cần số lượng đường đi đen trắng phải là số chẵn, do đó không tồn tại cách vẽ thêm cạnh nào thỏa mãn.
- Với test ví dụ số 3:



- Cách 1:
Có 3 đường đi đen trắng, là {1}, {2}, {3}.



- Cách 2:
Có 3 đường đi đen trắng, là {1}, {2}, {3}
- Với 2 test ví dụ còn lại, là để bạn đọc kiểm tra thuật toán của mình đã chính xác hay không.
- Test ví dụ số 5 sẽ không xuất hiện trong Large Dataset.

- **Bài toán sẽ được chia làm 2 Dataset:**

- Small Dataset: $1 \leq N \leq 5$.
- Large Dataset: $1 \leq N \leq 30$.

- **Giới hạn dữ liệu:**

- 5s / Dataset.
- 512Mb / Dataset.

-----Chúc các bạn làm bài tốt <3 <3 <3 -----