Contest: <https://oj.vnoi.info/contest/ndq_bitmask>

Phép dịch trái ( << ) : Với các vị trí bit khi dịch trái tràn ra ngoài thanh ghi => các bit bị loại bỏ. Nên thực chất (x << y) = (x << y) %

Solution

1. Bitwise:

* Làm bình thường. Chú ý các phép dịch trái phải ép kiểu unsigned int long long (VD: (1ULL << y))

1. Bò đen bò trắng

* Với mỗi vị trí (mỗi cột), lưu định dạng vị trí đó của con bò đen và con bò trắng
* Định dạng là một dãy bit, trong đó vị trí thứ 0/1/2/3 của dãy bit = 1 nếu tồn tại kí tự A/C/G/T trong dãy, bằng 0 nếu ngược lại
* Để kiểm tra bò trắng và bò đen trùng nhau không thì chỉ cần (bit đen & bit trắng) == 0.
* Code: https://ideone.com/j9PerD

1. Dãy chặn bit

* DP Bitmask : F(S): số hoán vị của tập S thỏa mãn điều kiện chặn bit
* Nhận xét: Nếu dãy a[1],a[2],..,a[k] bị chặn bit thì dãy a[1],a[2],..,a[k-1] bị chặn bit
* Công thức: F(S) = tổng các F( S \ {i} ) với {i} là các hoán vị thỏa mãn điều kiện chặt bit, phần tử cuối cùng của dãy hoán vị là i (i thuộc tập S)
* Vdu: S = {1,2,3} => F( {1,2,3} ) = F( {2,3} ) (bỏ 1 ở cuối) + F( {1,3} ) (bỏ 2 ở cuối) + F( {1,2} ) (bỏ 3 ở cuối)
* Ở đây thực chất là lấy các hoán vị thỏa mãn mà không có i, rồi thử ghép i vào cuối hoán vị
* n <= 20. => có 2^20 trạng thái (chọn và không chọn i trong tập S)
* Với ns là biểu diễn bit của tập S ( bit thứ k = 1 nếu phần tử k+1 có trong S, ngược lại = 0) => F(ns) = tổng các F( ns – ( 1<<(i-1) ) ) với mọi i thuộc tập S và thỏa mãn điều kiện chặn bit , tức [ (ns >> (i-1) ) & 1] == 1 và các phần tử khác thuộc tập S không có phần tử j nào mà (j&i) == i)
* Vdu: S = {2,4} => ns = 1010
* Đặt P(i) : {j : j&i = i, j <= n}
* Vdu: P(3) = {7,15,11}
* Đặt npi là biểu diễn bit của P(i) tương tự như S
* Để i thỏa mãn điều kiện chặn bit : S giao P(i) = rỗng => ns&npi == 0 (tất nhiên i phải thuộc S như trên)
* Cách cài đặt:

for(ns : 0 -> -1)

for(i : 1 -> n)

if( (i thuộc ns) && (ns & p[i]) == 0 ) F(ns) += F(ns – (1 << (i-1)))

* Trường hợp cơ bản: F(0) = 1
* Đáp án sẽ là F( (1 << i) – 1)
* Code: <https://ideone.com/dM13Ve>

1. Đếm hoán vị

* Nhận xét rằng -1 mọi phần tử trong hoán vị không ảnh hưởng kết quả đề bài, đưa đề bài về tính số hoán vị của tập 0,1,2,...,n-1 thỏa mãn 2 số liên tiếp nhau hơn kém nhau k đơn vị
* F(mask,j) : số dãy hoán vị thỏa mãn yêu cầu đề bài mà phần tử cuối của hoán vị là j
* Công thức: F(mask,i) = tổng các F(mask – (1 << i),j) với mọi j thỏa mãn bit thứ j trong mask là 1, j != i và |i-j| >= k
* Trường hợp cơ bản: F( (1 << i), i) = 1 với mọi 0 <= i <= n-1
* Đáp án sẽ là tổng các F( (1 << n)-1 ,i ) với mọi 0 <= i <= n-1
* Code: https://ideone.com/O33kGL

1. Ghép đôi

* Hàm đếm số lượng bit 1 của 1 dãy bit

+ Với số 32 bit: \_\_builtin\_popcount(bit)

+ Với số 64 bit: \_\_builtin\_popcountll(bit)

* F(mask) = số cách ghép các bạn nam trong mask với |mask| bạn nữ đầu tiên (kí hiệu |mask| là số lượng bit 1 trong mask, có thể làm như này vì số bạn nữ luôn bằng số bạn nam)
* Công thức = F(mask) = tổng các F(mask – (1 << (i-1))) với mọi i thỏa mãn bit thứ i-1 của mask là 1, và a[ i ][ |mask| ] == 1
* Trường hợp cơ bản: F(0) = 1
* Đáp án sẽ là F( (1<<n) – 1 )
* Code: <https://ideone.com/xKZ1gU>

Contest: https://oj.vnoi.info/contest/ndq\_sos

1. Sum over subset

* k = log(n) + 1
* Duyệt qua các submask của mask:

+ VD: submask của 1011 là 1011,1000,1001,1010,0000,0001,0010,0011

+ G(x): số bit 1 của x => x có submask

+ F(x) = tổng a[i] với mọi i : x&i=i, tức i là submask của x

+ Trường hợp cơ bản: F(i) = a[i]

+ for i : 0 -> k-1

For mask: 0 ->

If(mask >> i & 1)

F(mask) += F(mask – (1 << i))

+ Như vậy F(mask) sẽ được tính bởi các F(i) với i là tất cả submask của mask, và nó sẽ không bị tính trùng do mình for i trước

+ Đáp án là F(1),F(2),...,F( (1 << k) – 1 )

+ DPT: O( \* k)

* Code: <https://ideone.com/JfSg4d>

1. Sum over subset 2

* Y hệt bài trên, chỉ thay đổi là mình thay thành F[submask] += F[mask]

1. Bit problem

* k = max(log(a[i]) + 1)
* Cnt(v): số thằng có a[i] = v
* F(x) = tổng các cnt(i) với mọi i & x != 0

= n – cnt(i) với mọi i & x = 0

* Do mọi với mọi bit 1 của i thì bit tương ứng của x = 0 => (~x) = 1
* Vậy i & x = 0 => i & (~x) = i, ~x = x
* Bài toán đưa về tìm G(x) = tổng các cnt(i) với mọi i thỏa mãn i & (~x) = i