Bài 1: Chia nhóm

- Đây là một bài toán có dạng phát hiện chu trình của dãy số, ta hãy quan sát quá trình chia nhóm với $n = 5$, những đứa trẻ được chia vào nhóm theo thứ tự:
  1 2 3 4 5 4 3 2 | 1 2 3 4 5 4 3 2 | 1 2 3 4 5 4 3 2 | 1 2 3…
  Như ta có thể thấy trong ví dụ trên, trong quá trình chia nhóm, dãy số miêu tả thứ tự các nhóm được phân sẽ tạo thành một dãy có chu trình, chu trình này chính là:
  $$1, 2, 3, \dots, n, n-1, n-2, \dots, 2$$
  Dễ thấy rằng chu trình này gồm $2n-2$ phần tử. Nếu gọi a_i là số trẻ được phân vào nhóm i, khi đó với mỗi chu trình này, số lượng trẻ trong mỗi nhóm sẽ thay đổi như sau:
  $$a_1 += k$$
  $$a_n += k$$
  $$a_i += 2k, i = 2..(n-1)$$

- Như vậy ta chỉ cần đếm số lượng chu trình $cycle = m/(k*(2n-2))$. Khi đó ta có thể dễ dàng tính được mảng a_i sau $cycle$ chu trình trong $O(2n)$. Lượng trẻ còn lại không đủ cho một chu trình, vì vậy ta sẽ chia nhóm theo đúng miêu tả của bài toán với độ phức tạp là $O(2n)$.

- Vậy tổng độ phức tạp của thuật toán là $O(4n) \sim O(n)$.

Bài 2: Đếm hoán vị

- Để dễ quan sát hơn ta viết lại điều kiện của bài toán: một hoán vị của tập $\{1,2,..n\}$ gọi là dốc nếu như tồn tại một vị trí $1 < i < n$ sao cho:
  $$p_1 < p_2 < \cdots < p_i > p_{i+1} > p_{i+2} > \cdots > p_n$$

- Ta gọi phần tử p_i là đỉnh của hoán vị. Mấu chốt của bài toán đó là nhận xét $p_i = n$. Khi ta đã biết giá trị của p_i việc xây dựng hoán vị dốc là rất đơn giản. Ta xét lần lượt các phần tử còn lại 1, 2, …, n-1. Rõ ràng mỗi phần tử chỉ có 1 trong 2 trường hợp đó là nằm bên trái phần tử đỉnh hoặc nằm bên phải phần tử đỉnh. Sau khi đã phân hoạch được các phần tử còn lại vào bên trái và bên phải phần tử đỉnh, ta dễ dàng thu được hoán vị dốc bằng các phép sắp xếp.
- Như vậy, mỗi cách phân hoạch $n-1$ phần tử về bên trái hay bên phải phần tử đỉnh cho ta một hoán vị dốc. Mà số lượng cách phân hoạch này chính là số lượng dãy nhị phân có độ dài $n-1$. Tuy nhiên ta phải loại đi 2 trường hợp khi ở bên trái hoặc bên phải phần tử đỉnh p_i không có bất cứ phần tử nào (vì $1 < i < n$).

  Vậy kết quả của bài toán là: $2^{n-1} - 2$.
- Vì số n rất lớn nên ta cần tính kết quả bằng thuật toán chia để trị với độ phức tạp $O(logn)$ như sau:
  - Nếu n chẵn thì $2^n = 2^{\frac{n}{2}} * 2^{\frac{n}{2}}$
  - Nếu n lẻ thì $2^n = 2^{\frac{n}{2}} * 2^{\frac{n}{2}} * 2$
  - Như vậy để tính 2^n ta gọi đệ quy tính $2^{\frac{n}{2}}$. Neo đệ quy tại điểm $n = 0$

```
long long pow(long long a, long long n)
{
    if (n == 0) return 1;
    ll t = pow(a,n/2);
    t = (t * t) % base;
    if (n % 2) return (t * a) % base;
    else return t;
}
```

Bài 3: Thiên long bát bộ

- Nếu gọi x là số kỹ năng đạt cấp độ tối đa, y là cấp độ của kỹ năng đạt cấp độ thấp nhất, khi đó ta cần tìm x, y sao cho hàm số:
$$f(x, y) = cf * x + cm * y$$
đạt giá trị lớn nhất. Khoảng giá trị của các biến:
$$0 \leq x \leq n,$$
$$a_{\min} \leq y \leq a_{min} + m$$

- **Thuật toán 1 (30%):**
    - Như vậy ta có tất cả $n.m$ bộ giá trị (x, y) thỏa mãn ràng thuộc các biến. Như vậy thuật toán đơn giản nhất là thử tất cả $n.m$ bộ giá trị này và tìm giá trị hàm f lớn nhất.
    - Tuy nhiên ta cần kiểm tra xem giá trị (x, y) có là một giá trị "chấp nhận được" hay không?
    - Để kiểm tra điều này ta sẽ sử dụng thuật toán tham lam. Trước tiên sắp xếp lại dãy a_i theo thứ tự giảm dần. Sau đó ta sẽ thử tăng x kĩ năng có cấp độ cao nhất lên cấp độ A, tiếp theo tăng mọi kĩ năng còn lại lên cấp độ y. Phương án (x, y) là chấp nhận được nếu số lượng điểm kĩ năng m là đủ để thực hiện việc tăng cấp độ theo thuật toán tham lam trên. Như vậy để kiểm tra một phương án (x, y) có chấp nhận được hay không, ta sử dụng thuật toán có độ phức tạp là $O(n)$.
    - Vậy tổng độ phức tạp của thuật toán là $O(n^2m)$.

- **Thuật toán 2 (30%):**
    - Khi số m lớn thì khoảng giá trị của biến y càng lớn, khi đó thuật toán 1 không còn hiệu quả, ta cần sử dụng một thuật toán hiệu quả hơn để tìm cực đại hàm f.
    - Trước hết ta cũng sắp xếp mảng a theo thứ tự giảm dần như trong thuật toán 1. Gọi $sum_i = a_1 + a_2 + \cdots + a_i$ là tổng cấp độ của các kĩ năng từ 1 đến i.
    - Ở thuật toán này ta vẫn xét mọi giá trị có thể của x, khi đó số lượng điểm kĩ năng cần dùng để tăng x kĩ năng lên cấp độ tối đa có thể dễ dàng được tính theo công thức:
    $$m_1 = x * A - sum_x$$
    - Thay vì việc xét tất cả các giá trị của y như trong thuật toán 1, với mỗi giá trị x, ta sẽ xét từng phần tử của mảng a từ vị trí $x + 1$ đến n. Giả sử vị trí đang xét là vị trí c $(x < c \leq n)$. Nếu ta muốn biến y đạt giá trị ít nhất là a_c thì trước hết ta

cần tăng các kĩ năng $a_{c+1}, \dots, a_n$ lên cấp độ a_c, việc này tiêu tốn số lượng điểm kĩ năng là:

$$m_2 = a_c * (n - c) - (sum_n - sum_c)$$

Nếu giá trị $m - m_1 - m_2 \geq 0$, thì **lập tức dừng vòng lặp** c, lúc này biến y sẽ có giá trị $y = a_c + \frac{m - m_1 - m_2}{n - c + 1}$. Giá trị y này chính là giá trị làm cho hàm f đạt cực đại với giá trị x đang xét.

- Như vậy ta thu được thuật toán cực đại hàm f với độ phức tạp $O(n^2)$.

- **Thuật toán 3 (40%):**
  - Ta tiếp tục cải tiến bước xác định y trong thuật toán 2. Ở thuật toán 2 với mỗi giá trị của x ta cần tìm giá trị c nhỏ nhất có thể. Như vậy ta có thể sử dụng thuật toán tìm kiếm nhị phân để tìm giá trị c.
  - Kết hợp thuật toán 2 với thuật toán tìm kiếm nhị phân ta thu được thuật toán cuối cùng với độ phức tạp $O(nlogn)$.

------------------------- *Hết* -------------------------