

LỜI GIẢI ĐỀ THI THỬ LẦN 3

Tác giả: Mai Vinh Hiền, [Ta Chí Thành Danh](#) (tác giả lời giải bài Stick Lengths trên USACO Guide).

Bài 1 - MATHSKILLZ

Bài này chỉ cần đơn giản in ra số N và D cặp “00” đằng sau là ăn trọn điểm. Vì sao ư? Thử chứng minh đi :P

Vì các lần thi thử trước các bạn làm bài chán quá nên anh cho luôn một bài **free điểm** đấy. **Đề thi thật sẽ không cho bài 1 dễ thế này đâu!**

Bài 2 - MATHSKILLZAGAIN

Bài tương tự: [CSES - Stick Lengths \(Lời giải chi tiết\)](#)

Ta có hướng tiếp cận bài toán này như sau:

- Thử chọn một số nguyên k rất bé, ta nhận thấy ngay hàm f sẽ rất lớn.
- Thử chọn một số nguyên k rất lớn, ta nhận thấy hàm f cũng sẽ rất lớn.
- Thử chọn trung bình cộng của 2 giá trị trên, ta nhận thấy hàm f sẽ bé hơn cả 2 giá trị vừa trên.
- Từ đó ta có thể đoán rằng **hàm f sẽ có đồ thị hình lõm** (giống hàm số bậc 2 với hệ số a dương), có **điểm cực tiểu là k** bằng giá trị trung bình nào đó.

Và sự thật đúng là như vậy! (Nếu các bạn chứng minh toán học được thì càng tốt, nhưng vì hàm f ở đây rất đơn giản nên có thể tưởng tượng được dễ dàng)

Từ đó ta đặt ra câu hỏi “làm sao để tìm k sao cho đồ thị hàm số đạt giá trị tại **điểm cực tiểu** đó?”.

Đầu tiên ta rút gọn hàm f bằng cách tạo một dãy B sao cho $B_i = A_i - i$. Từ đó hàm f sẽ trở thành $f(A, k) = |B_1 - k| + |B_2 - k| + \dots + |B_n - k|$.

Khi đó hàm f đạt giá trị nhỏ nhất khi **k là số trung vị của dãy B**, tức phần tử thứ $n/2$ khi dãy B đã được sắp xếp tăng dần (bất kể n là số chẵn hay số lẻ vì **kết quả phải là số nguyên**).

Dù không biết khái niệm số trung vị thì vẫn có thể thử lấy k bằng $a[n/2]$, $a[n/2 + 1]$ hoặc $\text{int}((a[n/2] + a[n/2 + 1])/2)$ rồi lấy giá trị nhỏ nhất của hàm f tại mỗi k và ra kết quả đúng.

Vậy ta đã có ý tưởng cho thuật toán chuẩn của bài này, và ta cũng nhận ra một điều rằng muốn học Toán tốt thì phải có khả năng tưởng tượng tốt ;)

Bài 3 - NOTMATHSKILLZLMAO

Nguồn: Đề thi tuyển sinh vào lớp 10 trường Phổ Thông Năng Khiếu (2019-2020).

Ta có thể duyệt toàn bộ những đoạn có k phần tử để tìm ra kết quả, độ phức tạp của thuật toán là $O(N^2)$ với mỗi thao tác tìm ước chung lớn nhất của một đoạn con là $O(N)$. Code tham khảo:

```
1  #include <iostream>
2  #include <algorithm>
3
4  using namespace std;
5
6  #define gcd(a, b) abs(__gcd(a, b))
7
8  int main() {
9      int N, K; cin >> N >> K;
10     long long a[N]; for (long long x : a) cin >> x;
11
12     long long ans = LLONG_MIN;
13     for (int i = 0; i <= N - K; i++) {
14         long long res = a[i];
15         for (int j = i + 1; j < i + K; ++j) res = gcd(res, a[j]);
16         ans = max(ans, res);
17     }
18
19     cout << ans;
20 }
21
```

Còn thuật toán chuẩn thì ta phải cài đặt cấu trúc dữ liệu Segment Tree để lấy được ước chung lớn nhất của một đoạn con trong $O(\log N)$. Thuật toán của chúng ta sẽ có độ phức tạp là $O(N \log N)$. Code tham khảo: <https://ideone.com/mBs5fY> (code của một thầy bên THPT Gia Định).

Chứng minh thuật toán chuẩn của bài 2: [algorithm - Minimise the sum of difference between each element of an Array and an integer K - Stack Overflow](#) :p