

BATON

- Để độ bất ổn thấp ta tham lam lấy từ cột cao nhất bỏ sang cột bé nhất.
 - Vì thực hiện tối đa K thao tác nên sau mỗi lần thao tác ta phải cập nhật đáp án tối ưu
 - Cách làm đơn giản nhất đạt 4.3đ: Thực hiện k lần, mỗi lần sort lại rồi tăng $a[1]$ và giảm $a[n]$
 - Cách làm full điểm: Chỉ sort một lần duy nhất ở đầu, Thực hiện K lần: Tăng $a[1]$, giảm $a[n]$ sau đó hoán đổi $a[1]$ và $a[n]$ đến vị trí cần thiết.
- Ví dụ: dãy 1, 1, 2, 2, 4, 4 → sau khi tăng $a[1]$ và giảm $a[n]$ ta có 2,1,2,2,4,3 vị trí ta thấy không cần sort lại mà chỉ cần dùng vòng while để đưa số 2 ở vị trí 1 và số 3 ở cuối vào đúng vị trí → 1, 2,2,2,3,4
- Bài này có một số sai rất đáng tiếc: khai báo mảng 1000, sau mỗi thao tác không cập nhật đáp án.

SHORTPATH

- Là một bài xử lý xâu bình thường không có gì phải bàn. Làm theo đúng yêu cầu đề bài. Có bạn không đọc kỹ đề bài nên xử lý thiếu.

STRPRIME

- Bài này khó xử lý nếu theo hướng cắt xâu ra mà tìm. Cắt ra không biết số đó dài hay ngắn, chuyển thành số có được không?
- Cực kỳ dễ xử lý nếu làm ngược lại, tức là tìm xem có số nguyên tố x trong xâu hay không bằng hàm pos.
- Vì đề bài yêu cầu chỉ tìm các số nguyên tố chưa đến 10^5 nên ta dùng eratos sinh ra các số nguyên tố cho vào một mảng trước. Sau đó duyệt các số nguyên tố vừa tạo ra rồi kiểm tra xem nó có mặt trong xâu hay không bằng hàm pos.

TDPRIME

- Bài này nắm vững kỹ thuật phân tích một số ra thừa số nguyên tố là OK.
- Cách phân tích dễ đúng và dễ cài đặt là sử dụng vòng for: Duyệt i từ 2 đến \sqrt{n} : Nếu n chia hết cho i thì chứng tỏ i là thừa số nguyên tố, khi đó đưa i vào đáp án và chia n cho i

```
For i:=2 to sqrt(n) do
  If n mod i = 0 then Begin
    - Đưa i vào danh sách thừa số
    - While n mod i = 0 do begin
      o  $n := n \div i$ ;
      o Đếm thừa số i xuất hiện bao nhiêu lần
    - End;
  End;
```

If ($n > 1$) then đưa n vào danh sách thừa số

- Bài này ít test nhưng toàn test đẹp nên code không chuẩn không full được. Một số bạn bị TLE hai test cuối. Một số bạn code sai. Một số dùng lệnh while $n > 1$ do ($i=i+1$) để cài đặt sẽ bị TLE khi n là một số nguyên tố lớn. Khi đó i chạy đến n thì nó mới dừng nên độ phức tạp thành ra $O(n)$.