

## دورنگی

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

تئوری "4 رنگ آمیزی" میگوید که هر نقشه ای رو میشه با 4 تا رنگ، رنگ آمیزی کرد به طوری که هر دو منطقه ای که همسایه ی هم باشن، رنگ های متفاوتی داشته باشن

الان شما قراره که مسئله ی ساده تری رو حل کنین.

به شما یک گراف داده میشه و شما باید تصمیم بگیرید که آیا میشه راس های این گراف را با استفاده از تنها 2 رنگ جوری رنگ آمیزی کرد که هیچ دو راسی که همسایه اند (یعنی توسط یک یال به یکدیگر متصل اند) هم رنگ نباشن

برای ساده سازی مسئله میتونین فرض های زیر رو در نظر بگیرید 1. هیچ راسی طوقه (یال به خودش) ندارد 2. گراف جهت دار نیست. یعنی اگر گفته شد که راس a از طریق یال ab به b متصل است شما باید فرض کنید که راس b نیز توسط یال ab به a متصل است 3. گراف همبند است. یعنی از هر راس به تمام راس های دیگر مسیر وجود دارد

## ورودی

ورودی شامل تعداد زیادی تست کیس است. برای هر تست کیس ابتدا عدد  $n < 200$  که بیانگر تعداد راس هاست داده میشود. هر راس توسط یک عدد از 0 تا  $n - 1$  نام گذاری شده است. در خط بعدی مقدار L داده میشود که بیانگر تعداد یال هاست. در L خط بعدی، در هر خط دو عدد داده میشود که بیانگر دو راسی است که به یکدیگر متصل اند. ورودی با مقدار  $n = 0$  بیانگر پایان برنامه است

## خروجی

شما باید تصمیم بگیرید که میتوان این گراف را با دو رنگ، رنگ آمیزی کرد یا خیر و مانند مثال چاپ کنید

## ورودی نمونه

```
3
3
0 1
1 2
2 0
3
2
0 1
1 2
9
8
0 1
0 2
0 3
0 4
0 5
0 6
0 7
0 8
0
```

## خروجی نمونه

NOT BICOLORABLE.  
BICOLORABLE.  
BICOLORABLE.

## مرتب سازی سریع

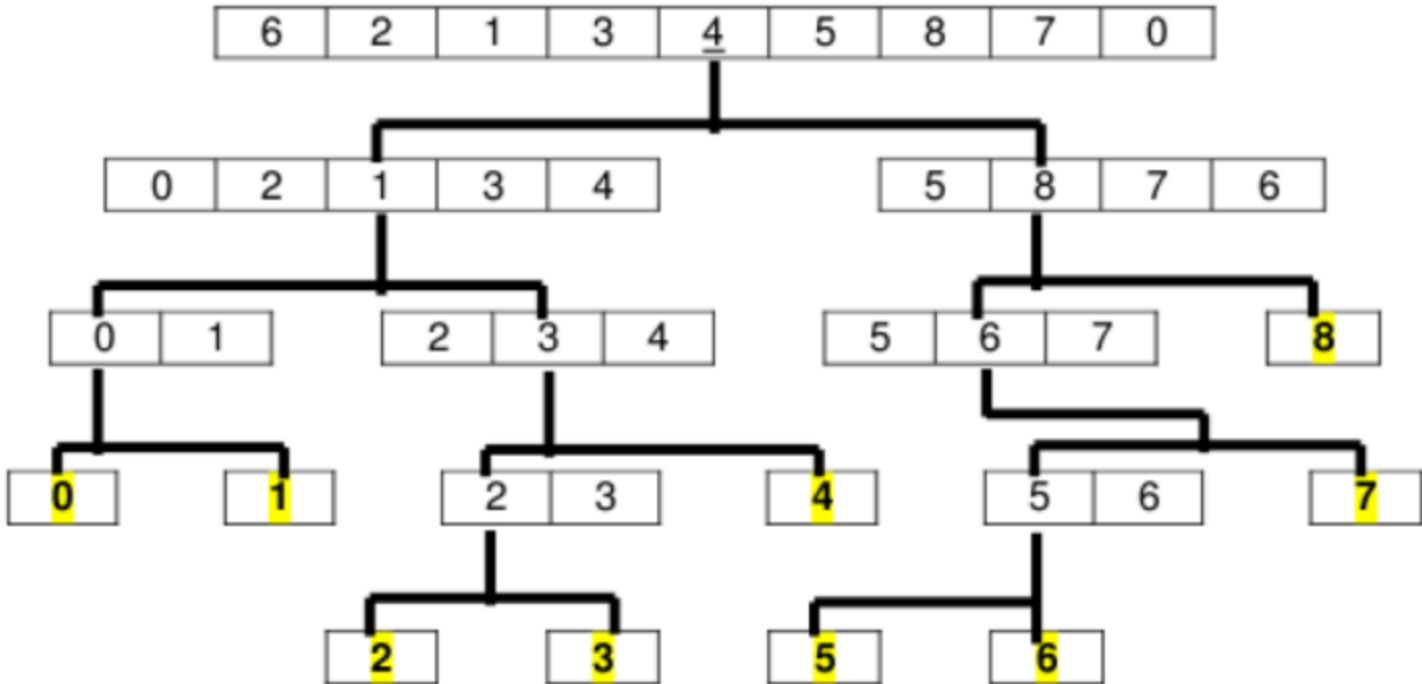
- محدودیت زمان: 2 ثانیه
- محدودیت حافظه: 128 مگابایت

مرتب سازی سریع از جمله روش‌های محبوب و با سرعت بالای برای مرتب کردن داده‌ها محسوب می‌شود. این روش هم مثل روش ادغامی از الگوریتم تقسیم و حل برای مرتب کردن داده‌ها استفاده می‌کند. به این ترتیب که داده‌ها را به دو قسمت مجزا تقسیم و با مرتب‌کردن آن‌ها کل داده‌ها را مرتب می‌کند. برای اینکار یکی از داده‌ها(مثلا داده اول) به عنوان محور انتخاب می‌شود. داده‌ها براساس محور طوری چینش می‌شوند که همه داده‌های کوچکتر از محور، سمت چپ و داده‌های بزرگتر یا مساوی با آن در سمت راستش قرار می‌گیرند. با مرتب کردن دو قسمت به دست آمده کل داده‌ها مرتب می‌شوند. در این حالت مثل روش ادغام نیازی به ادغام کردن داده‌ها نیست، چرا که قسمت سمت راست همگی از قسمت سمت چپ کوچکتر هستند و بالعکس. مثلاً اعداد صحیح زیر را در نظر بگیرید:

5 61 9 0 4 5 15 3 1 14 10

1 0 4 3 1 4 5 6 9 5 15 10

همانطور که مشاهده می‌کنید اعداد سمت چپ عدد 5 همگی از 5 کوچکتر اعداد سمت راست همگی بزرگتر یا مساوی آن هستند. در تصویر زیر چگونگی انجام این الگوریتم نشان داده شده است. با این تفاوت که محور الگوریتم، عنصر وسط در نظر گرفته شده است:



برنامه‌ای بنویسید که یک رشته از کاربر دریافت کند و سپس رشته مرتب شده را چاپ کند.

### ورودی

در تنها خط ورودی یک رشته از اعداد به شما داده می‌شود. اعداد از ۱۰۰۰۰۰۰۰۰۰۰ کوچکترند.

همچنین تعداد اعداد از ۵۰۰۰۰۰ کمتر است.

### خروجی

در تنها خط خروجی اعداد را به صورت مرتب شده صعودی چاپ کنید.

### مثال

#### نمونه ورودی

0 1 5 3 6 2 4

نمونه خروجی

0 1 2 3 4 5 6