

جایگشت مفید

- محدودیت زمان: ۲ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

احتمالا از پیش با الگوریتم Merge Sort آشنا هستید. امین برای اینکه این الگوریتم را کمی بهینه کند، تصمیم گرفته تنها در صورتی که آرایه از پیش مرتب شده نباشد، آن را به دو نیم تقسیم کرده و برای هر کدام از دو قسمت، تابع mergeSort را صدا بزند. برای مثال آرایه $[1, 2, 3, 4, 5]$ را در نظر بگیرید. ابتدا یک بار تابع mergeSort را برای این آرایه صدا می‌کنیم. این تابع با بررسی آرایه متوجه می‌شود که این آرایه در همین حالت مرتب شده است و در همین نقطه، از تابع خارج می‌شود. در واقع در این حالت **فقط یک بار** این تابع صدا زده شده است. حال آرایه $[3, 2, 5]$ را در نظر بگیرید. ابتدا یک بار تابع mergeSort را برای این آرایه صدا می‌زنیم. این تابع آرایه را بررسی می‌کند و متوجه می‌شود که آرایه از قبل به صورت مرتب شده نیست. در نتیجه، اندیس mid را به صورت زیر پیدا می‌کند:

$$mid = \left\lfloor \frac{l + r}{2} \right\rfloor = \left\lfloor \frac{0 + 3}{2} \right\rfloor = 1$$

سپس، یک بار تابع mergeSort را برای اندیس 0 تا 1 (آرایه $[3]$) و بار دیگر برای اندیس 1 تا 3 (آرایه $[2, 5]$) صدا می‌زند. همانطور که مثال‌ها نشان می‌دهد، آرایه‌ها شامل اندیس انتهایی نیستند. در نهایت می‌بینیم که به ازای آرایه $[3, 2, 5]$ ، تابع mergeSort به تعداد 3 بار صدا زده شده است. حال امین برای توسعه الگوریتم دیگری، نیاز دارد یک جایگشت مفید با اندازه n پیدا کند؛ به طوری که اگر بخواهیم الگوریتم Merge Sort را بر روی آن اعمال کنیم، دقیقا m بار تابع mergeSort بر روی آن صدا زده شود. یک جایگشت مفید با اندازه n یک آرایه n عضوی است که شامل تمام اعداد بازه $[1, n]$ خواهد بود.

ورودی

ورودی تنها شامل یک خط است که در آن به ترتیب دو عدد n و m که توسط فاصله از هم جدا شده‌اند، داده می‌شود.

$$1 \leq n \leq 10^5$$

$$1 \leq m \leq 2 \times 10^5$$

خروجی

خروجی شامل یک خط است که در آن، n عدد که توسط فاصله از هم جدا شده‌اند، که اعضای جایگشت مفید با اندازه n خواهد بود. همچنین در صورتی که چنین جایگشتی وجود ندارد، مقدار خروجی 1- خواهد بود.

مثال

ورودی نمونه 1

3 3

خروجی نمونه 1

2 1 3

ورودی نمونه 2

4 1

خروجی نمونه 2

1 2 3 4

ورودی نمونه 3

5 6

خروجی نمونه 3

-1