

square869120Contest#1

E問題 散歩

by E869120, square1001

問題概要

- N 個の街と $N-1$ 本の道路があり,街 i にはそれぞれ整数 $a[i]$ が書かれている。道路 i は街 i と街 $i+1$ を結んでいて,長さは $a[i]^a[i+1]$ である。
- 街1から出発したくさんの街を順番に通って街1に戻るような経路のうち最短のもの長さを $\text{mod } 1,000,000,007$ で求めなさい。

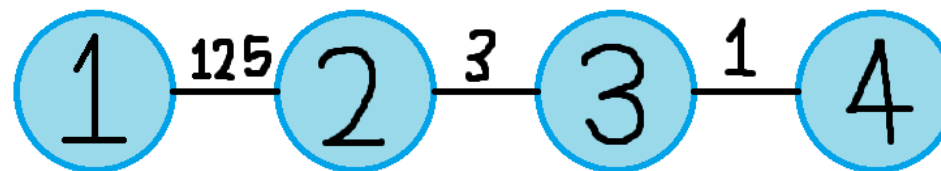
問題概要

- 制約
 - $2 \leq N \leq 120,000$
 - $1 \leq Q \leq 120,000$
 - $1 \leq a[i] \leq 1,000,000,000$

具体例

- $N = 4, Q = 3, a = \{ 5, 3, 1, 2 \}, c = \{ 3, 2, 4 \}$ のとき

- 街1-街2を結ぶ道路の長さは125
- 街2-街3を結ぶ道路の長さは3
- 街3-街4を結ぶ道路の長さは1

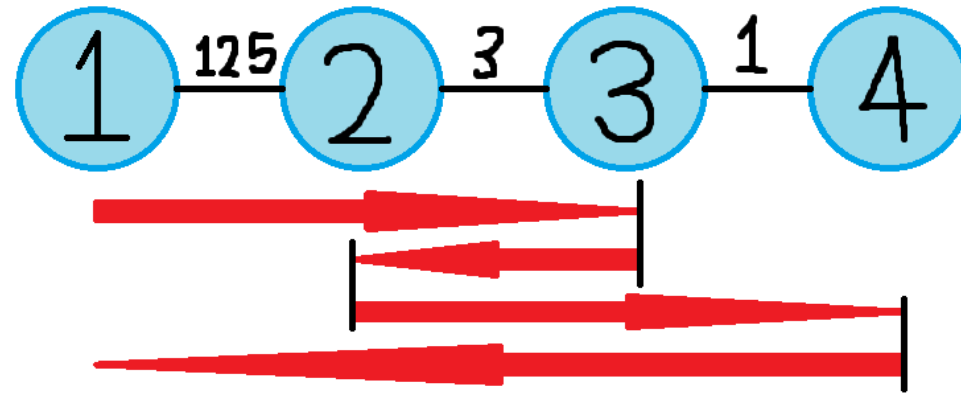


具体例

- 右のような経路で行くと,

$$(125+3)+(3)+(3+1)+(1+3+125)$$

= 264 となる。



部分点解法(30点)

- シミュレーションする

- 街 i から街 j に行く経路の長さは,

- $a[i]^a[i+1]+a[i+1]^a[i+2]+\dots+a[j-1]^a[j]$ で表される。

- それを $Q+1$ 回行う。

⇒オーバーフローに注意し, mod 1,000,000,007していくと, 正しい答えが得られる。

⇒計算量は $O(NQ \cdot \max(a[i]))$ なので最大で1,000,000,000程度となる。よって, 間に合わない可能性がある。

部分点解法(30点)

- 問題点

- $a[i]^a[i+1]$ を求めることが同じ i に対して何回も実行されている。

⇒すべての i ($1 \leq i \leq N-1$) に対して最初に $a[i]^a[i+1] \bmod 1,000,000,007$ を求めておく。

⇒計算量は $O(N \cdot \max(a[i]) + NQ)$ であるので最大でも2,000,000程度となり間に合う。

部分点解法(60点)

- 制約
 - $1 \leq N \leq 1,000$
 - $1 \leq Q \leq 1,000$
 - $1 \leq a[i] \leq 1,000,000,000$ <-- ここが30点と違う！

⇒ $a[i]^a[i+1]$ を求めるのを高速化しなければならない。

部分点解法(60点)

- $a[i]^a[i+1]$ を求めるのを高速化する方法
 - 繰り返し自乗法を使う。

```
int modpow(int a, int b){
    if(b==0){return 1;}
    int res=modpow(a,b/2);
    res=(long long)res*res%mod;
    return res%2==1?res*a:res;
}
```

部分点解法(60点)

- 繰り返し自乗法を使うことで、 $a^b \bmod 1,000,000,007$ が $O(\log b)$ で求められる。

⇒ $O(\sum \log(a[i]) + NQ)$ によって最大でも1,030,000程度なので間に合う。

満点解法

- 制約
 - $1 \leq N \leq 120,000$
 - $1 \leq Q \leq 120,000$
 - $1 \leq a[i] \leq 1,000,000,000$

⇒シミュレーションに時間をかけないようにしたい。

満点解法

-
- $b[i]=a[i]^a[i+1]$ とすると, $i \rightarrow j (i < j)$ に行くときの距離は $b[i]+b[i+1]+\dots+b[j-1]$ で求められる。
 - それを求めるのに $O(j-i)$ かかっているので, それを高速化する方法はないか?

満点解法

- 累積和を使う
 - $d[i]=b[1]+b[2]+\dots+b[i]$ を記録する
 - $d[1]=b[1]$
 - $d[i]=d[i-1]+b[i]$ ($i \geq 2$)
 - そのようにすると, $b[i]+b[i+1]+\dots+b[j-1]$ は $d[j-1]-b[i-1]$ で求められる。(modには注意)
- ⇒計算量は $O(\sum \log(a[i]) + Q)$ となり,最大でも4,000,000程度となる。よって間に合う。