Wild operations (wild)

Filippo, Francesco'nun çılgın dizi işlemlerini yönetme yeteneğini test etmek istiyor, bu yüzden ona N uzunluğunda bir dizi olan $A_0, ..., A_{N-1}$'i verdi.

Şimdi Filippo, Francesco'dan dizi üzerinde bazı işlemler yapmasını isteyecek. Her işlem şöyle olabilir:

- Bir x tam sayısı ve geçerli bir p indeksi için A_p'nin değerini x olarak *değiştirme*.
- $[l, r]$ aralığını *bozma* (perturb), yani tüm $l < p \leq r$ için $A_p = \max(A_p, A_{p-1})$ değerini **eş zamanlı olarak** ayarlama.

Herhangi bir anda Filippo, Francesco'dan geçerli bir p indeksi için A_p'nin değerini söylemesini isteyebilir.

Francesco çok meşgul, bu yüzden Filippo'nun sorularını yanıtlamak için senden yardım istemeye karar verdi.

Implementasyon

`.cpp` uzantılı tek bir dosya göndermelisin.

> ⇦ Bu görev için ekler arasında, örnek bir implementasyon içeren bir `wild.cpp` şablonu bulacaksın.

Aşağıdaki fonksiyonları kodlamalısın:

C++	`void init(int N, vector<int> A);`

- Bu fonksiyon, programının çalışmasının başında bir kez çağrılır.
- N tam sayısı, dizinin uzunluğudur.
- 0'dan $N-1$'e kadar indekslenmiş A dizisi, Filippo'nun seçtiği başlangıç dizisidir.

C++	`void change(int p, int x);`

- Bu fonksiyon, programının çalışması sırasında Filippo bir değişiklik yaptığında birçok kez çağrılır.
- p tam sayısı, dizideki değiştirilen değerin indeksidir.
- x tam sayısı, atanacak yeni değerdir.

C++	`void perturb(int l, int r);`

- Bu fonksiyon, programının çalışması sırasında Filippo bir aralığı bozduğunda birçok kez çağrılır.
- l tam sayısı, Filippo'nun bozduğu aralığın sol ucudur.
- r tam sayısı, Filippo'nun bozduğu aralığın sağ ucudur.

C++	`int calc(int p);`

- Bu fonksiyon, programının çalışması sırasında Filippo bir dizi elemanının değerini sorduğunda birçok kez çağrılır.

- p tam sayısı, Filippo'nun sorduğu elemanın indeksidir.
- Fonksiyon, önceki tüm işlemler uygulandıktan sonra A_p'nin değerini dönmelidir.

Örnek Değerlendirici

Düzeltme sırasında kullanılan değerlendiricinin (grader) basitleştirilmiş bir versiyonu, bu problemle ilgili dizinde mevcut. Çözümlerini yerel olarak test etmek için kullanabilirsin. Örnek değerlendirici, girdi verilerini `stdin`'dan okur, implemente etmen gereken fonksiyonu çağırır ve aşağıdaki formatta `stdout`'a yazar.

Filippo tarafından yapılan; değişiklik, bozma ve soru işlemlerinin toplam sayısını Q olarak kabul edelim.
O zaman, girdi dosyası $2 + Q$ satırdan oluşur ve şunları içerir:

- Satır 1: N, Q tam sayıları.
- Satır 2: N adet $A_0, ..., A_{N-1}$ tam sayısı, dizinin başlangıç değerleri.
- Satır $3 + i$ ($0 \leq i < Q$): 2 veya 3 tam sayı, aşağıdaki formatlardan birinde:
  - 1 p x: Filippo'nun A_p'yi x olarak değiştirmesi anlamına gelir.
  - 2 l r: Filippo'nun $[l, r]$ aralığını bozması anlamına gelir;
  - 3 p: Filippo'nun A_p'nin değerini sorması anlamına gelir.

Çıktı dosyası, `calc` fonksiyonu tarafından dönen değerleri içeren Q_3 satırdan (burada Q_3, `calc` çağrılarının sayısıdır) oluşur.

Kısıtlamalar

- $1 \leq N \leq 400\,000$.
- $0 \leq Q \leq 400\,000$.
- $1 \leq A_i \leq 10^9$ tüm $0 \leq i < N$ için.
- $0 \leq p < N$ her `change` ve `calc` çağrısında.
- $0 \leq l < r \leq N - 1$ her `perturb` çağrısında.
- $1 \leq x \leq 10^9$ her `change` çağrısında.

Puanlama

Programın, alt görevlere gruplandırılmış birkaç test durumu üzerinde test edilecek. Bir alt görev için puan almak için, içindeki tüm test durumlarını doğru bir şekilde çözmelisin.

Bir test durumunda `change` fonksiyonuna yapılan çağrı sayısını Q_1 olarak kabul edelim, o zaman:

- **Alt görev 0 [0 puan]**: Örnek.
- **Alt görev 1 [15 puan]**: `change` fonksiyonu asla çağrılmaz; her `perturb` çağrısında $l = 0$, $r = N - 1$.
- **Alt görev 2 [16 puan]**: Tüm $0 \leq i < N$ için $A_i \leq 10$ ve `change`e yapılan tüm çağrılar için $x \leq 10$.
- **Alt görev 3 [13 puan]**: `change` fonksiyonuna yapılan çağrılar değerleri azaltmaz ($x \geq A_p$), $Q_1 \leq 1000$ ve her `perturb` çağrısında $l = 0$, $r = N - 1$.
- **Alt görev 4 [22 puan]**: `change` fonksiyonu asla çağrılmaz.
- **Alt görev 5 [14 puan]**: `change` fonksiyonuna yapılan çağrılar değerleri azaltmaz ($x \geq A_p$), $Q_1 \leq 1000$.
- **Alt görev 6 [20 puan]**: Ek kısıtlama yok.

Örnekler

stdin	stdout
10 28 5 1 7 8 3 2 5 6 9 4 1 1 1 1 0 1 2 0 1 2 2 6 1 6 5 2 2 9 2 2 5 2 4 5 1 4 5 2 3 8 1 8 4 3 0 1 6 3 1 4 1 2 5 7 1 0 3 2 4 5 1 6 3 3 0 3 1 3 2 3 3 3 4 3 5 3 6 3 7 3 8 3 9	1 3 1 7 8 1 8 3 6 4 9

Açıklama

$A = [5, 1, 7, 8, 3, 2, 5, 6, 9, 4]$ dizisiyle başlıyoruz.

- Olay 1: Filippo A_1'i 1 olarak değiştiriyor (zaten 1 idi): yeni dizi $[5, 1, 7, 8, 3, 2, 5, 6, 9, 4]$.
- Olay 2: Filippo A_0'ı 1 olarak değiştiriyor: yeni dizi $[1, 1, 7, 8, 3, 2, 5, 6, 9, 4]$.
- Olay 3: Filippo $[0, 1]$'i bozuyor: yeni dizi $[1, 1, 7, 8, 3, 2, 5, 6, 9, 4]$.
- Olay 4: Filippo $[2, 6]$'yı bozuyor: yeni dizi $[1, 1, 7, 8, 8, 3, 5, 6, 9, 4]$.

Olay 19′dan itibaren, Filippo sadece dizideki değerleri soruyor, değişiklik veya bozma işlemi yapmıyor. Bu noktada dizi $[3, 1, 7, 8, 1, 8, 3, 6, 4, 9]$ şeklindedir.