messy
Country: MNG

Алдаа илрүүлэх

Илшат бол үр ашигтай өгөгдлийн бүтэц дээр ажилладаг програм хангамжийн инженер юм. Нэг өдөр тэр шинэ өгөгдлийн бүтэц зохиосон байна. Энэ өгөгдлийн бүтэц нь *сөрөг бус*, n-битийн бүхэл тоонуудын олонлогийг хадгалж чадна. Энд n нь хоёрын зэрэг юм. Өөрөөр хэлбэл, ямар нэг сөрөг бус бүхэл тоо b-ийн хувьд $n = 2^b$ байна.

Уг өгөгдлийн бүтэц нь анх хоосон байна. Энэ өгөгдлийн бүтцийг хэрэгжүүлж байгаа програм нь дараах дүрмийг баримтална:

- Програм нь n-битийн бүхэл тоон элементүүдийг өгөгдлийн бүтэц рүү нэг нэгээр нь `add_element(x)` функцийг ашиглан нэмж чадна. Хэрэв програм нь өгөгдлийн бүтцэд байгаа элементийг нэмэхийг оролдвол юу ч өөрчлөгдөхгүй.
- Програм нь хамгийн сүүлийн элементээ нэмсний дараа `compile_set()` функцийг яг нэг удаа дуудна.
- Эцэст нь програм , x элементийг өгөгдлийн бүтцэд байгаа эсэхийг шалгахын тулд `check_element(x)` функцийг дуудаж болно. Энэ функцийг хэд хэдэн удаа хэрэглэж болно.

Илшат анх энэ өгөгдлийн бүтцийг хэрэгжүүлэхдээ `compile_set()` функцийг алдаатай бичсэн. Энэ алдаанаас болж элемент бүрийн хоёртын цифрүүдийн дараалал нь ижил маягаар өөрчлөгддөг. Илшат танаас энэ алдаанаас болж цифрүүдийн дараалал яг ямар байдлаар өөрчлөгддөгийг олж өгөхийг хүссэн.

0-ээс $n-1$ хүртлэх тоонууд яг нэг удаа орсон $p = [p_0, \ldots, p_{n-1}]$ дарааллыг авч үзье. Бид ийм дарааллыг *сэлгэмэл* гэж нэрлэдэг. Хоёртын цифрүүд нь $a_0, \ldots, a_{n-1}$ байх (a_0 нь хамгийн ахлах орон) олонлогийн нэг элементийг авч үзье. `compile_set()` функц дуудагдахад энэ элемент нь $a_{p_0}, a_{p_1}, \ldots, a_{p_{n-1}}$ элементээр солигдоно.

p гэсэн ганц сэлгэмлийг бүх элементүүдийн цифрүүдийн дарааллыг өөрчлөхөд хэрэглэнэ. Уг сэлгэмэл нь дурын сэлгэмэл байж болох ба $0 \le i \le n-1$ нөхцлийг хангах i бүрийн хувьд $p_i = i$ байх сэлгэмэл ч байж болно.

Жишээ нь $n = 4$, $p = [2, 1, 3, 0]$ бөгөөд та олонлог руу `0000`, `1100` болон `0111` гэсэн хоёртын бичлэгтэй тоонуудыг нэмсэн гэж үзье. `compile_set` функцийг дуудсанаар эдгээр элементүүдийг харгалзан `0000`, `0101` ба `1110` болгон өөрчилнө.

Таны даалгавар бол өгөгдлийн бүтэцтэй харилцах замаар p сэлгэмлийг олох явдал юм. Энэ нь (дараах дарааллаар хийнэ):

1. n -битийн бүхэл тоонуудыг сонгон авна,
2. эдгээр бүхэл тоонуудыг өгөгдлийн бүтэц рүү хийнэ,
3. `compile_set` функцийг дуудаж алдааг хийлгэнэ,
4. өөрчлөгдсөн олонлогт зарим элементүүд байгаа эсэхийг шалгана,
5. эдгээр мэдээллийг хэрэглэн p сэлгэмлийг тодорхойлж олоод буцаана.

Таны програм `compile_set` функцийг зөвхөн нэг удаа дуудаж болно гэдгийг анхаар.

Мөн таны програм сангийн функцуудыг дуудах тоо хязгаартай байна. Тэдгээр нь
- `add_element` функцийг дээд тал нь w удаа дуудна (w нь "writes" гэсэн үг),
- `check_element` функцийг дээд тал нь r удаа дуудна (r нь "reads" гэсэн үг).

Хэрэгжүүлэлтийн мэдээлэл

Та нэг функц (арга) хэрэгжүүлнэ:
- `int[] restore_permutation(int n, int w, int r)`
  - `n`: олонлогийн элемент бүрийн хоёртын дүрслэл дэх битийн тоо (мөн p -ийн урт).
  - `w`: таны програмын `add_element` үйлдлийг гүйцэтгэж болох хамгийн их тоо.
  - `r`: таны програмын `check_element` үйлдлийг гүйцэтгэж болох хамгийн их тоо.
  - уг функц нь сэргээсэн p сэлгэмлээ буцаана.

C хэлэн дээр уг функцийн тодорхойлолт нь арай өөр байна:
- `void restore_permutation(int n, int w, int r, int* result)`
  - `n`, `w` болон `r` нь өмнөхтэй ижил утгатай.
  - уг функц нь сэргээсэн p сэлгэмлээ өгөгдсөн `result` массивт бичиж хадгална: i утга бүрийн хувьд p_i -г `result[i]` руу бичнэ.

Сангийн функцууд

Өгөгдлийн бүтэцтэй харилцахын тулд таны програм доорх гурван функцийг (аргыг) хэрэглэнэ:
- `void add_element(string x)`
  Энэ функц нь `x` гэж тодорхойлогдсон элементийг олонлогт нэмнэ.
  - `x`: олонлогт нэмэх гэж байгаа бүхэл тооны хоёртын дүрслэлийг илэрхийлэх `'0'` болон `'1'`-ээс тогтсон тэмдэгт мөр. `x`-ийн урт нь n байх ёстой.

- `void compile_set()`
  Энэ функцийг яг нэг удаа дуудах ёстой. Таны програм энэ дуудалтын дараа `add_element()` функцийг дуудаж болохгүй. Таны програм энэ дуудалтаас өмнө `check_element()` функцийг дуудаж болохгүй.
- `boolean check_element(string x)`
  Энэ функц нь өөрчлөгдсөн олонлогт `x` элемент байгаа эсэхийг шалгана.
  - `x`: шалгах гэж байгаа элементийн хоёртын дүрслэлийг илэрхийлэх `'0'` болон `'1'`-ээс тогтох тэмдэгт мөр. `x`-ийн урт нь n байх ёстой.
  - хэрэв `x` элемент өөрчлөгдсөн олонлогт байгаа бол `true`-г буцаах ба

байхгүй бол `false`-г буцаана.

Хэрэв таны програм дээрх хязгаарлалтуудын аль нэгийг зөрчвөл шалгагчийн хариу нь "Wrong Answer" байх болно.

Бүх тэмдэгт мөрийн хувьд эхний тэмдэгт нь харгалзах бүхэл тооны хамгийн ахлах орон байна.

Шалгагч нь The grader fixes the permutation p before the function `restore_permutation` is called.

Өөрийн програмчлалын хэл дээрх хэрэгжүүлэлтийн талаарх дэлгэрэнгүй мэдээллийг авахын тулд өгөгдсөн загвар файлуудыг хэрэглээрэй.

Жишээ

Шалгагч нь дараах функцийн дуудалтыг хийнэ:

- `restore_permutation(4, 16, 16)`. $n = 4$ ба програм дээд тал нь 16 "writes" болон 16 "reads" хийж чадна.

Програм нь доорх функцийн дуудалтуудыг хийнэ:

- `add_element("0001")`
- `add_element("0011")`
- `add_element("0100")`
- `compile_set()`
- `check_element("0001")` нь `false`-г буцаана
- `check_element("0010")` нь `true`-г буцаана
- `check_element("0100")` нь `true`-г буцаана
- `check_element("1000")` нь `false`-г буцаана
- `check_element("0011")` нь `false`-г буцаана
- `check_element("0101")` нь `false`-г буцаана
- `check_element("1001")` нь `false`-г буцаана
- `check_element("0110")` нь `false`-г буцаана
- `check_element("1010")` нь `true`-г буцаана
- `check_element("1100")` нь `false`-г буцаана

`check_element()` функцийн эдгээр буцаасан утгуудтай ганц л сэлгэмэл тохирно: $p = [2, 1, 3, 0]$ сэлгэмэл. Иймд, `restore_permutation` функц `[2, 1, 3, 0]`-г буцаана.

Дэд бодлогууд

1. (20 оноо) $n = 8$, $w = 256$, $r = 256$, дээд тал нь 2 ширхэг i индексийн хувьд $p_i \neq i$ байна ($0 \leq i \leq n - 1$),
2. (18 оноо) $n = 32$, $w = 320$, $r = 1024$,
3. (11 оноо) $n = 32$, $w = 1024$, $r = 320$,
4. (21 оноо) $n = 128$, $w = 1792$, $r = 1792$,
5. (30 оноо) $n = 128$, $w = 896$, $r = 896$.

Жишээ шалгагч

Жишээ шалгагч нь оролтыг доорх форматаар уншина:

- мөр 1: n, w, r бүхэл тоонууд,
- мөр 2: p-гийн элементүүдийг илэрхийлэх n ширхэг бүхэл тоо.