



## Харааны програм

Та роботод харааны програм бичиж өгөх гэж байна. Робот камераар зураг авах болгонд хар цагаанаар санах ойдоо хадгалдаг. Зураг болгон нь 0-ээс  $H - 1$  хүртэл дугаарлагдсан мөр болон 0-ээс  $W - 1$  хүртэл дугаарлагдсан багануудаас бүрдэх  $H \times W$  хэмжээтэй пикселүүдээс бүрдэх тор юм. Энд **яг хоёр ширхэг** хар пиксел байгаа бөгөөд бусад бүх пиксел нь цагаан байх юм.

Робот нь энгийн хэдэн үйлдлүүдээс бүтэх програмыг ашиглаад зураг болгоныг боловсруулдаг. Танд эерэг бүхэл  $H$ ,  $W$  болон  $K$  тоонууд өгөгджээ. Таны зорилго бол ямар ч зургийн хувьд хоёр хар пикселүүдийн хоорондох **зай** нь яг  $K$  эсэхийг тодорхойлдог программыг үүсгэж өгөх юм. Энд  $r_1$  мөрний  $c_1$  багана дах болон  $r_2$  мөрний  $c_2$  багана дах пикселүүдийн хоорондох зай нь  $|r_1 - r_2| + |c_1 - c_2|$  байна. Энэхүү томъёондох  $|x|$  нь  $x$  утгын абсолют утгыг хэлэх буюу хэрэв  $x < 0$  бол  $-x$ , хэрэв  $x \geq 0$  бол  $x$  байх юм.

Бид одоо робот яаж ажилладгийг тайлбарлах болно.

Роботын санах ой нь хангалттай том бөгөөд 0-ээс эхлэн дугаарлагдсан нүднүүдээс тогтоно. Нүд болгон нь 0 эсвэл 1 утгыг авах бөгөөд утга олгогдсон бол хэзээ ч өөрчлөгдөхгүй юм. Зураг нь мөр, мөрөөр 0-ээс  $H \cdot W - 1$  хүртэл дугаарлагдсан нүднүүдэд хадгалагдана. Эхний мөрний пикселүүд 0-ээс  $W - 1$  нүднүүдэд хадгалагдах бөгөөд сүүлийн мөрний пикселүүд  $(H - 1) \cdot W$ -с  $H \cdot W - 1$  хүртэлх нүднүүдэд хадгалагдах юм. Тухайлбал  $i$ -р мөрний  $j$ -р баганы пиксел нь  $i \cdot W + j$  нүдэнд хадгалагдах бөгөөд хар байвал 1, үгүй бол 0 утгыг авна.

Роботын програм нь 0-ээс эхлэн дараалсан тоонуудаар дугаарлагдсан **үйлдлүүдийн** дараалал юм. Програм эхлэхэд энэхүү үйлдлүүд нь нэг нэгээрээ ажиллах юм. Үйлдэл болгон нь нэг эсвэл олон нүднүүдийн утгыг унших (эдгээр утгуудыг үйлдлийн **оролт** гэе) бөгөөд 0 эсвэл 1 утгыг гаргаж ирэх юм (энэхүү утгыг үйлдлийн **гаралт** гэе).  $i$ -р үйлдлийн гаралт нь  $H \cdot W + i$  нүдэнд хадгалагдана.  $i$ -р үйлдлийн оролт нь зөвхөн пикселүүд болон өмнөх үйлдлүүдийн гаралтуудыг ашиглаж болно. Өөрөөр хэлбэл 0-ээс  $H \cdot W + i - 1$  нүднүүдийг ашиглаж болох юм.

Энд дөрвөн төрлийн үйлдлүүд байгаа:

- NOT: яг ганц оролттой. Үүний гаралт нь хэрэв оролт нь 0 байвал 1, үгүй бол 0 байна.
- AND: нэг эсвэл олон оролттой. Үүний гаралт нь хэрэв **бүх** оролт нь 1 байвал 1

байна.

- OR: нэг эсвэл олон оролттой. Үүний гаралт нь хэрэв **ядаж нэг** оролт нь 1 байвал 1 байна.
- XOR: нэг эсвэл олон оролттой. Үүний гаралт нь хэрэв оролтод **сондгой тооны** 1 байвал 1 байна.

Таны програмын сүүлийн үйлдлийн гаралт нь хэрэв хоёр хар пикселүүдийн хоорондох зай нь яг  $K$  байвал 1 байх ёстой, үгүй бол 0 байх ёстой юм.

## Хэрэгжүүлэлтийн мэдээлэл

Та дараах функцуудыг хэрэгжүүлэх ёстой:

```
void construct_network(int H, int W, int K)
```

- $H, W$ : роботын камераас авсан зургийн өндөр болон урт
- $K$ : бүхэл тоо
- Энэхүү функц нь роботын програмыг гаргаж ирэх ёстой. Робот ямар ч зураг авсан бай энэхүү програм нь хоёр хар пикселүүдийн хоорондох зай яг  $K$  байгаа эсэхийг тодорхойлдог байх ёстой.

Энэхүү функц нь дараах үйлдлүүдийг роботын програмд (анхандаа хоосон байгаа) үйлдэл залгахын тулд дуудаж болно.

```
int add_not(int N)
int add_and(int[] Ns)
int add_or(int[] Ns)
int add_xor(int[] Ns)
```

- Харгалзан NOT, AND, OR, or XOR үйлдлүүдийг залгана.
- $N$  (add\_not функцийн хувьд): залгагдаж буй NOT үйлдлийн уншиж байгаа оролтын нүдний дугаар
- $Ns$  (add\_and, add\_or, add\_xor функцүүдийн хувьд): залгагдаж буй AND, OR, эсвэл XOR үйлдлийн оролтоор уншиж буй нүднүүдийн дугаарыг агуулах массив
- Функц болгон гаралтын хадгалагдсан нүдний дугаарыг буцаана. Энэхүү функцийн дараалсан дуудалт нь  $H \cdot W$  тооноос эхлэн дараалсан тоонуудыг буцаах болно.

Роботын програм хамгийн ихдээ 10 000 үйлдлээс тогтож болно. Энэхүү үйлдлүүд нь хамгийн ихдээ 1 000 000 нүднүүдийг уншиж болно. Өөрөөр хэлбэл add\_and, add\_or болон add\_xor функцүүдийн  $Ns$  массивуудын хэмжээний нийлбэр дээр add\_not функцийн дуудалтын тоог нэмэхэд 1 000 000-с хэтрэхгүй байх ёстой гэсэн үг юм.

Бүх үйлдлүүдийг залгасны дараа construct\_network функц дуусах ёстой. Роботын

программыг хэд хэдэн тооны зургууд дээр ажиллууж үзнэ. Таны бодолтыг хэрэв өгөгдсөн тестийн зураг бүрийн хувьд сүүлийн үйлдлийн гаралт 1 байх зайлшгүй бөгөөд хүрэлцээтэй нөхцөл нь хоёр хар пиксел хоорондох зай нь яг  $K$  байдаг бол тэнцсэн гэж үзнэ.

Шалгагч нь таны бодолтонд дараах алдааны мессэжүүдийг гаргаж болох юм:

The grading of your solution may result in one of the following error messages:

- Instruction with no inputs: `add_and`, `add_or`, эсвэл `add_xor` функцэд хоосон массив оруулсан үед.
- Invalid index: `add_and`, `add_or`, `add_xor`, эсвэл `add_not` функцэд буруу нүдний дугаар (магадгүй сөрөг) оруулсан үед.
- Too many instructions: таны програм 10 000-аас их үйлдэл залгахыг оролдсон үед.
- Too many inputs: үйлдлүүд нь нийтдээ 1 000 000-аас олон утга уншсан үед.

## Жишээ

$H = 2$ ,  $W = 3$ ,  $K = 3$  байг. Тэгвэл энд хоёр хар пикселүүдийн хоорондох зай нь яг 3 байдаг хоёр л зураг байгаа.

0	1	2
3	4	5

0	1	2
3	4	5

- Тохиолдол 1: хар пикселүүд нь 0 болон 5
- Тохиолдол 2: хар пикселүүд нь 2 болон 3

Боломжит хариулт нь дараах үйлдлүүдийг роботын програмд залгах юм:

1. `add_and([0, 5])` Энэхүү үйлдлийн гаралт нь 1 байх зайлшгүй бөгөөд хүрэлцээтэй нөхцөл нь тохиолдол 1 юм. Гаралт нь 6 дугаартай нүдэнд хадгалагдана.
2. `add_and([2, 3])` Энэхүү үйлдлийн гаралт нь 1 байх зайлшгүй бөгөөд хүрэлцээтэй нөхцөл нь тохиолдол 2 юм. Гаралт нь 7 дугаартай нүдэнд хадгалагдана.
3. `add_and([6, 7])` Энэхүү үйлдлийн гаралт нь 1 байх зайлшгүй бөгөөд хүрэлцээтэй нөхцөл нь дээрх тохиолдын аль нэг нь байх юм.

## Хязгаарлалтууд

- $1 \leq H \leq 200$
- $1 \leq W \leq 200$

- $2 \leq H \cdot W$
- $1 \leq K \leq H + W - 2$

## Дэд бодлогууд

1. (10 оноо)  $\max(H, W) \leq 3$
2. (11 оноо)  $\max(H, W) \leq 10$
3. (11 оноо)  $\max(H, W) \leq 30$
4. (15 оноо)  $\max(H, W) \leq 100$
5. (12 оноо)  $\min(H, W) = 1$
6. (8 оноо) Зураг болгоны хувьд 0-р мөрний 0-р баганы пиксел заавал хар байна.
7. (14 оноо)  $K = 1$
8. (19 оноо) Нэмэлт хязгаарлалт байхгүй.

## Жишээ шалгагч

Жишээ шалгагч нь дараах байдлаар оролтыг уншина:

- мөр 1:  $H \ W \ K$
- мөр  $2 + i$  ( $i \geq 0$ ):  $r_1[i] \ c_1[i] \ r_2[i] \ c_2[i]$
- сүүлийн мөр:  $-1$

Эхний болон сүүлийн мөрөөс бусад мөр нь хоёр хар пикселийг агуулах зургийг илэрхийлнэ. Бид  $2 + i$  мөрөнд байгаа зургийг  $i$ -р зураг байна гэж тэмдэглэж байгаа. Нэг хар пиксел нь  $r_1$  мөрний  $c_1$  баганы нүдэнд нөгөө нь  $r_2$  мөрний  $c_2$  баганы нүдэнд байгаа гэж үзнэ.

Жишээ шалгагч нь эхлээд `construct_network(H, W, K)` функцийг дуудна. Хэрэв `construct_network` функц нь бодлогын өгүүлбэр дээрх дурьдсан зарим хязгаарлалтыг зөрчвөл жишээ шалгагч нь хэрэгжүүлэлтийн мэдээллийн төгсгөл дэх алдааны мэдээллийн аль нэгийг хэвлэх болно.

Үгүй бол жишээ шалгагч нь дараах хоёр гаралтыг үүсгэнэ.

Нэгдүгээрт, жишээ шалгагч нь роботын програмын гаралтыг дараах байдлаар хэвлэнэ:

- мөр  $1 + i$  ( $0 \leq i$ ):  $i$ -р зургийн хувьд роботын програмын сүүлчийн үйлдлийн гаралт (1 эсвэл 0).

Хоёрдугаарт, жишээ шалгагч нь тухайн газарт `log.txt` файлыг дараах хэлбэртэйгээр үүсгэнэ:

- мөр  $1 + i$  ( $0 \leq i$ ):  $m[i][0] \ m[i][1] \ \dots \ m[i][c - 1]$

$i + 1$ -р мөрөнд байгаа дараалал нь роботод  $i$ -р зургийг даруулаад программыг

ажиллуулаад ямар утга роботын санах ойд хадгалагдсан байгааг илэрхийлнэ. Тухайлбал  $m[i][j]$  нь  $j$ -р нүдний утгыг агуулна.  $s$ -н утга (дарааллын урт) нь  $H \cdot W$  дээр роботын үйлдлүүдийн тоог нэмсэнтэй тэнцүү гэдгийг анхаарна уу.