

## Ангараг

Фараонууд анх сансарт гарсан гэдгийг хүн бүр мэддэг билээ. Тэд Тутмус I гариг (одоо Ангараг гэж нэрлэгддэг) дээр хөл тавихын тулд анхны сансрын хөлгийг хөөргөсөн. Гаригийн гадаргуу нь  $(2n + 1) \times (2n + 1)$  хэмжээтэй, хуурай газар эсвэл ус байж болох квадрат үүрүүдээс тогтоно.  $i$ -р мөрийн  $j$ -р багана ( $0 \leq i, j \leq 2 \cdot n$ ) дээр байгаа үүрийн хувьд хуурай газар бол  $s[i][j] = '1'$  харин ус байгаа бол  $s[i][j] = '0'$  байна.

Хэрэв хоёр хуурай газрын үүрийн хувьд тэдгээрийг холбосон, дараалсан хоёр үүр бүр нь ерөнхий талтай байх, хуурай газрын үүрүүдээс тогтсон зам байвал тэр хоёр үүрийг холбогдсон байна гэж хэлнэ. Гариг дээрх арал гэдэг нь аль ч хоёр үүр нь холбогдсон байх хуурай газрын үүрүүдийн максимал олонлог байна.

Сансрын хөлгийн даалгавар нь гариг дээрх арлын тоог олох явдал юм. Гэвч уг даалгавар нь хөлгийн эртний компьютерийн хувьд хялбар ажил биш байв. Компьютер нь  $h$  гэсэн санах ойтой ба тэнд өгөгдлийг  $(2n + 1) \times (2n + 1)$  гэсэн хоёр хэмжээтэй массивт хадгална. Энд массивын элемент бүр нь 100-аас хэтрэхгүй урттай, тэмдэгт бүр нь '0' (ASCII 48) эсвэл '1' (ASCII 49) утгатай байх хоёртын тэмдэгт мөрийг хадгалж чадна. Анх санах ойн үүр бүрийн эхний бит нь үүрүүдийн төлвийг хадгална:  $h[i][j][0] = s[i][j]$  (бүх  $0 \leq i, j \leq 2 \cdot n$  утгын хувьд).

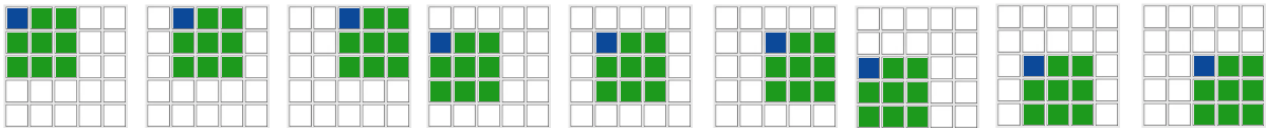
Санах ойд хадгалагдсан өгөгдлийг боловсруулахын тулд компьютер зөвхөн санах ойн  $3 \times 3$  хэмжээтэй муж руу хандаж, уг мужийн зүүн дээд талын үүрийн утгыг дарж бичнэ. Өөрөөр хэлбэл компьютер  $h[i..i + 2][j..j + 2]$  ( $0 \leq i, j \leq 2 \cdot (n - 1)$ ) мужид хандах ба  $h[i][j]$  байрлал дээрх утгыг дарж бичнэ. Энэ ажиллагааг үүнээс хойш  $(i, j)$  **үүрийг боловсруулах** гэж нэрлэнэ.

Компьютерийн хүчин чадлын хязгаарлалтаас болж Фараонууд дараах механизмыг санал болгосон:

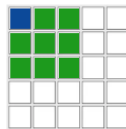
- Компьютер нь санах ойг  $n$  алхамтайгаар боловсруулна.
- $k$ -р ( $0 \leq k \leq n - 1$ ) алхамд  $m = 2 \cdot (n - k - 1)$  гэж үзвэл компьютер  $i$ -ийн өсөх дарааллаар,  $i$  бүрийн хувьд  $j$ -ийн өсөх дарааллаар cell  $(i, j)$  үүрийг ( $0 \leq i, j \leq m$ ) боловсруулна. Өөрөөр хэлбэл компьютер үүрүүдийг доорх дарааллаар боловсруулна:  $(0, 0), (0, 1), \dots, (0, m), (1, 0), (1, 1), \dots, (1, m), \dots, (m, 0), (m, 1), \dots, (m, m)$ .
- Сүүлийн алхамд ( $k = n - 1$ ) компьютер зөвхөн  $(0, 0)$  үүрийг л боловсруулна. Үүний дараа  $h[0][0]$  дахь утга нь гариг дээрх арлуудын тоотой тэнцүү болсон байна.

Доорх диаграм нь компьютер  $5 \times 5$  ( $n = 2$ ) хэмжээтэй санах ойг яаж боловсруулж байгааг харуулна. Цэнхэр үүр нь дарж бичиж байгаа үүрийг харуулах бол бусад өнгөтэй үүр нь боловсруулагдаж байгаа дэд массивыг илэрхийлнэ.

0-р алхмын үед компьютер доорх дэд массивуудыг зурагт үзүүлсэн дарааллаар боловсруулна:



1-р алхмын үед компьютер зөвхөн нэг дэд массивыг боловсруулна:



Таны даалгавар бол компьютер яаж ажилладаг нь өгөгдсөн үед Тутмус I гариг дээрх арлын тоог тооцоолох функцийг бичих явдал юм.

## Хэрэгжүүлэлтийн мэдээлэл

Та дараах функцийг хэрэгжүүлнэ:

```
int64 process(int64[][] a, int i, int j, int k, int n)
```

- $a$ : Боловсруулагдаж байгаа дэд массивыг илэрхийлэх  $3 \times 3$  хэмжээтэй массив, тухайлбал  $a = h[i..i+2][j..j+2]$ .
- $i, j$ : компьютерийн боловсруулж буй үүрийн мөр болон баганын дугаар.
- $k$ : одоогийн алхмын дугаар.
- $n$ : нийт алхмын тоо болон  $(2n+1) \times (2n+1)$  хэмжээтэй гаригийн гадаргуугийн хэмжээсийг илэрхийлэх тоо.
- Уг функц нь  $10^{15}$ -аас хэтрэхгүй, сөрөг бус бүхэл тоо буцаана. Буцаасан утга нь компьютерийн санах ойд  $h[i][j]$  байрлалд хадгалагдана.
- Уг функцийг хамгийн сүүлийн удаа  $k = n - 1$  үед дуудна. Энэ дуудалтын үед, уг функц нь гариг дээрх арлын тоог буцаана.
- Уг функц нь ямар ч статик болон глобал хувьсагчаас хамааралгүй байх ба буцаах утга нь зөвхөн дамжуулсан параметруудээс хамаарах ёстой.

Тест бүр бие даасан  $T$  ширхэг үзэгдлийг агуулна (өөрөөр хэлбэл ялгаатай гаригийн гадаргуу). Таны хэрэгжүүлэлтийн ажиллагаа нь үзэгдэл бүрийн хувьд үзэгдлийн дарааллаас хамаарахгүй байна. Учир нь `process` функцийг үзэгдэл бүр дээрх дуудалт нь заавал дараалсан байдлаар хийгдэхгүй байж болно. Гэвч, үзэгдэл бүрийн хувьд `process` функцийг дуудалтууд нь бодлогын өгүүлбэрт бичсэн дарааллаар хийгдэнэ.

Мөн, тест бүрийн хувьд таны програмыг тус тусд нь зэрэг ажиллуулж эхлүүлэх болно. Санах ой болон Төв процессорын хугацааны хязгаарлалтыг шалгахдаа эдгээр ажиллагаануудын хувьд тэдгээрийн нийлбэрийг ашиглана. Санаатайгаар эдгээр ажиллагаануудын хооронд өгөгдөл гадуур дамжуулахыг оролдвол хуурах оролдлого гэж үзэх ба хасагдах шалгтгаан болно.

Тухайлбал, `process` функцийг дуудалтын үед статик эсвэл глобал хувьсагч руу хадгалсан аливаа мэдээлэл нь дараагийн функцуудийн дуудалтын үед хадгалагдах эсэх нь баталгаагүй байна.

## Хязгаарлалт

- $1 \leq T \leq 10$
- $1 \leq n \leq 20$
- $s[i][j]$  нь '0'(ASCII 48) эсвэл '1'(ASCII 49) байна (бүх  $0 \leq i, j \leq 2 \cdot n$  утгын хувьд)
- $h[i][j]$ -ийн урт нь яг 100 байна (бүх  $0 \leq i, j \leq 2 \cdot n$  утгын хувьд)
- $h[i][j]$ -ийн тэмдэгт бүр нь '0' (ASCII 48) эсвэл '1' (ASCII 49) утгатай байна (бүх  $0 \leq i, j \leq 2 \cdot n$  утгын хувьд)

`process` функцийн дуудалт бүрийн хувьд:

- $0 \leq k \leq n - 1$
- $0 \leq i, j \leq 2 \cdot (n - k - 1)$

## Дэд бодлого

1. (6 оноо)  $n \leq 2$
2. (8 оноо)  $n \leq 4$
3. (7 оноо)  $n \leq 6$
4. (8 оноо)  $n \leq 8$
5. (7 оноо)  $n \leq 10$
6. (8 оноо)  $n \leq 12$
7. (10 оноо)  $n \leq 14$
8. (24 оноо)  $n \leq 16$
9. (11 оноо)  $n \leq 18$
10. (11 оноо)  $n \leq 20$

## Жишээ

### Жишээ 1

$n = 1$  ба  $s$  нь доорх байдалтай байх тохиолдлыг авч үзье:

```
1 0 0
1 1 0
0 0 1
```

Энэ жишээн дээр гаригийн гадаргуу нь  $3 \times 3$  үүрүүд ба 2 арлаас тогтоно. Энд `process` функцийг зөвхөн 1 удаа дуудна.

0-р алхмын үед шалгагч нь `process` функцийг яг нэг удаа дуудна:

```
process([["100", "000", "000"], ["100", "100", "000"], ["000", "000", "100"]], 0, 0, 0, 1)
```

$h$ -ийн үүр бүрийн зөвхөн эхний 3 битийг харуулсан байгааг анхаараарай.

Уг функцийн дуудалт нь "0100..." утгыг буцаана (орхисон битүүд нь бүгд тэгүүд). Энд ...0010 нь аравтын тооллын системд 2-той тэнцүү байна. 96 ширхэг тэгийг орхиж ...-ээр тэмдэглэсэн байгааг анхаараарай.

## Жишээ 2

$n = 2$  ба  $s$  нь доорх байдалтай байх тохиолдлыг авч үзье:

```
1 1 0 1 1
1 1 0 0 0
1 0 1 1 1
0 1 0 0 0
0 1 1 1 1
```

Энэ жишээн дээр гаригийн гадаргуу  $5 \times 5$  үүрүүдээс болон 4 арлаас тогтоно. Энд process функцийг дуудах 2 алхам байна.

0-р алхмын үед шалгагч нь process функцийг 9 удаа дуудна:

```
process(["100","100","000"],["100","100","000"],["100","000","100"],0,0,0,2)
process(["100","000","100"],["100","000","000"],["000","100","100"],0,1,0,2)
process(["000","100","100"],["000","000","000"],["100","100","100"],0,2,0,2)
process(["100","100","000"],["100","000","100"],["000","100","000"],1,0,0,2)
process(["100","000","000"],["000","100","100"],["100","000","000"],1,1,0,2)
process(["000","000","000"],["100","100","100"],["000","000","000"],1,2,0,2)
process(["100","000","100"],["000","100","000"],["000","100","100"],2,0,0,2)
process(["000","100","100"],["100","000","000"],["100","100","100"],2,1,0,2)
process(["100","100","100"],["000","000","000"],["100","100","100"],2,2,0,2)
```

Дээрх дуудалтууд харгалзан "011", "000", "000", "111", "111", "011", "110", "010", "111" утгуудыг буцаасан гэж үзье. Энд орхисон битүүд нь тэг байна. Тэгвэл 0-р алхам дууссаны дараа  $h$ -д дараах утгууд хадгалагдсан байна:

```
"011", "000", "000", "100", "100"
"111", "111", "011", "000", "000"
"110", "010", "111", "100", "100"
"000", "100", "000", "000", "000"
"000", "100", "100", "100", "100"
```

1-р алхмын үед шалгагч нь process функцийг нэг удаа дуудна:

```
process(["011","000","000"],["111","111","011"],["110","010","111"],0,0,1,2)
```

Уг функцийн дуудалт нь "0010000...." утгыг буцаана (орхисон битүүд нь бүгд тэгүүд). Энд ....0000100 нь аравтын тооллын системд 4-тэй тэнцүү байна. 93 ширхэг тэгийг орхиж ...-ээр тэмдэглэсэн байгааг анхаараарай.

## Жишээ шалгагч

Жишээ шалгагч оролтыг доорх форматаар уншина:

- мөр 1:  $T$
- блок  $i$  ( $0 \leq i \leq T - 1$ ):  $i$ -р үзэгдлийг илэрхийлэх блок.
  - мөр 1:  $n$
  - мөр  $2 + j$  ( $0 \leq j \leq 2 \cdot n$ ):  $s[j][0] \ s[j][1] \ \dots \ s[j][2 \cdot n]$

Жишээ шалгагч үр дүнг доорх форматаар хэвлэнэ:

- мөр  $1 + i$  ( $0 \leq i \leq T - 1$ ):  $i$ -р үзэгдлийн хувьд `process` функцийг буцаах сүүлийн утга.