

Mars

Հայտնի է, որ փարավոններն են առաջինը դուրս եկել տիեզերք: Նրանք ստեղծեցին առաջին տիեզերանավը և ոտք դրեցին Թուբմուս I (այժմ հայտնի է որպես Մարս) մոլորակի վրա: Մոլորակի մակերևույթը կարելի է մոդելավորել $(2n + 1) \times (2n + 1)$ չափի վանդակավոր դաշտի տեսքով, որի յուրաքանչյուր վանդակ կամ հող է, կամ ջուր: i -րդ տողի և j -րդ սյան ($0 \leq i, j \leq 2 \cdot n$) հատման վանդակը $s[i][j] = '1'$, եթե այն հող է, և $s[i][j] = '0'$, եթե այն ջուր է:

Կասենք, որ երկու հող պարունակող վանդակների կապակցված են, եթե գոյություն ունի նրանց միացնող հող պարունակող վանդակներից կազմված ճանապարհ, որի երկու հարևան վանդակներն ընդհանուր կող ունեն: Կղզի կոչվում է այն մաքսիմալ թվով վանդակների բազմությունը, որին պատկանող ցանկացած երկու վանդակ կապակցված են:

Տիեզերանավի խնդիրն էր պարզել մոլորակում կղզիների քանակը: Սակայն դա հեշտ խնդիր չէր տիեզերանավի հին համակարգչի պատճառով: Համակարգիչն ուներ h հիշողություն, որը պահում էր տվյալները երկու չափանի $(2n + 1) \times (2n + 1)$ չափի զանգվածի տեսքով, որտեղ զանգվածի յուրաքանչյուր տարրում կարելի է պահել մինչև 100 երկարության չգերզանցող '0' (ASCII 48) և '1' (ASCII 49) սիմվոլներից կազմված տող: Սկզբում յուրաքանչյուր վանդակի հիշողության առաջին բիթերում պահվում են վանդակավոր դաշտի վանդակների արժեքները. $h[i][j][0] = s[i][j]$ (բոլոր $0 \leq i, j \leq 2 \cdot n$ համար): h -ի մնացած բոլոր բիթերի նախնական արժեքները '0' (ASCII 48) են:

Հիշողությունում պահվող տվյալները մշակելու համար համակարգիչը կարող է դիմել միայն 3×3 չափի հիշողության մասի և փոխել այդ մասի վերևի ձախ վանդակի արժեքը: Ավելի ֆորմալ, համակարգիչը կարող է դիմել $h[i..i + 2][j..j + 2]$ ($0 \leq i, j \leq 2 \cdot (n - 1)$) տարրերին և վերագրում անել $h[i][j]$ -ին: Այս Այս գործընթացը հետագայում կանվանվի որպես **մշակել (i, j) վանդակը**.

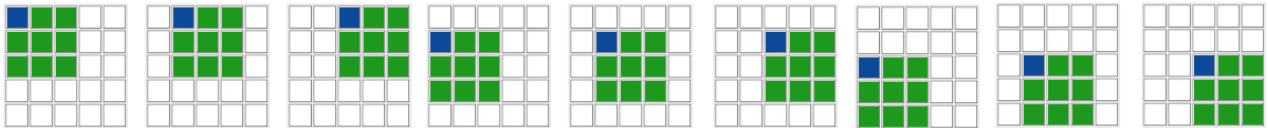
Հաղթահարելով համակարգչի սահմանափակումները՝ փարավոնները մշակեցին հետևյալ մեխանիզմը.

- Համակարգիչը կմշակի հիշողությունը n փուլերով:
- k -րդ ($0 \leq k \leq n - 1$) փուլին, թող $m = 2 \cdot (n - k - 1)$, համակարգիչը կմշակի (i, j) վանդակը բոլոր $0 \leq i, j \leq m$ համար, i -ի աճման կարգով, իսկ յուրաքանչյուր i համար, j -ի աճման կարգով: Այլ խոսքերով. համակարգիչը կմշակի վանդակները հետևյալ կարգով. $(0, 0), (0, 1), \dots, (0, m), (1, 0), (1, 1), \dots, (1, m), \dots, (m, 0), (m, 1), \dots, (m, m)$:
- Վերջին փուլում ($k = n - 1$), համակարգիչը կմշակի միայն $(0, 0)$ վանդակը: Որից հետո $h[0][0]$ -ում գրված արժեքը պետք է հավասար լինի մոլորակում կղզիների

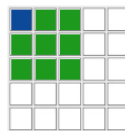
քանակին երկուական համակարգով, որտեղ թվի ամենափոքր նշանակալից բիթը պահվում է որպես տողի առաջին սիմվոլ:

Ստորև բերված նկարում ցույց է տրված, թե համակարգիչն ինչպես է մշակում 5×5 չափի հիշողությունը ($n = 2$): Կապույտ գույնով ցույց է տրված այն վանդակը, որի արժեքը փոխվում է, իսկ գունավոր վանդակները այն ենթազանգվածի վանդակներն են, որը մշակվում է:

0 փուլի ընթացքում համակարգիչը կմշակի ենթազանգվածները հետևյալ հերթականությամբ.



1 փուլի ժամանակ համակարգիչը կմշակի միայն մեկ ենթազանգված:



Ձեր խնդիրն է իրականացնել մեթոդ, որը թույլ կտա համակարգչին հաշվել կղզիների քանակը Թուրքիոս I մոլորակում հաշվի առնելով այն, թե ինչպես է այն գործում:

Իրականացման մանրամասներ

Դուք պետք է իրականացնեք հետևյալ ենթածրագիրը.

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

- a : a 3×3 զանգված, որը ցույց է տալիս տվյալ պահին մշակվող ենթազանգվածը, մասնավորապես, $a = h[i..i+2][j..j+2]$, որտեղ a -ի յուրաքանչյուր տարրը ճիշտ 100 երկարության տող է, և յուրաքանչյուր սիմվոլ կամ '0' (ASCII 48) է, կամ '1' (ASCII 49):
- i, j : այն վանդակի տողի և սյան համարը, որը համակարգիչը տվյալ պահին մշակում է:
- k : ընթացիկ փուլի համարը:
- n : փուլերի ընդհանուր քանակը, և մոլորակի մակերևույթի չափերը, որը բաղկացած է $(2n+1) \times (2n+1)$ վանդակներից:
- Այս ենթածրագիրը պետք է վերադարձնի 100 երկարության երկուական տող: Վերադարձվող արժեքը կպահվի համակարգչի հիշողությունում $h[i][j]$ տեղում:
- Այս ենթածրագրի վերջին կանչը կլինի այն ժամանակ, երբ $k = n - 1$: Այդ կանչի ժամանակ ենթածրագիրը պետք է վերադարձնի մոլորակում կղզիների քանակը երկուական ներկայացումով, որտեղ ամենափոքր նշանակալից բիթը ներկայացված է 0 ինդեքսում (տողի առաջին սիմվոլում), երկրորդ փոքր նշանակալից բիթի ինդեքսը 1 է, և այդպես շարունակ:

- Ենթածրագրում չպետք է կիրառվեն ստատիկ կամ գլոբալ փոփոխականներ, և նրա վերադարձի արժեքը պետք է կախված լինի միայն իրեն փոխանցված պարամետրերից:

Յուրաքանչյուր թեստ ներառում է T իրարից անկախ սցենարներ (այսինքն տարբեր մոլորակների մակերևույթներ): Ձեր իրականացումը պետք է կախված չլինի սցենարների հերթականությունից, քանի որ `process` ենթածրագրի կանչերը յուրաքանչյուր սցենարի համար պարտադիր չէ, որ հերթականությամբ արվեն: Սակայն յուրաքանչյուր սցենարի համար երաշխավորվում է, որ `process`-ի կանչերը տեղի կունենան խնդրում նշված կարգով:

Բացի այդ, յուրաքանչյուր թեստի համար ձեր ծրագրի մի շարք օրինակներ կսկսեն աշխատել միաժամանակ: Հիշողության և CPU ժամանակի սահմանափակումները բոլոր այդ օրինակների համար կոմբինացվում են: Այս օրինակների միջև տվյալների շրջանակից դուրս տվյալ փոխանցելու ցանկացած միտումնավոր փորձ համարվում է խաբեություն և որակագրկման պատճառ:

Մասնավորապես, չի երաշխավորվում, որ `process`-ի կանչի ժամանակ ստատիկ կամ գլոբալ փոփոխականներում պահվող ինֆորմացիան հաջորդ կանչի ժամանակ կարող է հասանելի լինել:

Սահմանափակումներ

- $1 \leq T \leq 10$
- $1 \leq n \leq 20$
- $s[i][j]$ -ն կամ '0' (ASCII 48) է, կամ '1' (ASCII 49) (բոլոր $0 \leq i, j \leq 2 \cdot n$ համար)
- $h[i][j]$ -ի երկարությունը ճիշտ 100 է (բոլոր $0 \leq i, j \leq 2 \cdot n$ համար)
- $h[i][j]$ -ի յուրաքանչյուր սիմվոլ '0' (ASCII 48) է կամ '1' (ASCII 49) (բոլոր $0 \leq i, j \leq 2 \cdot n$ համար)

`process` ենթածրագրի յուրաքանչյուր կանչի համար

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

Ենթախնդիրներ

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

10. (11 միավոր) $n \leq 20$

Օրինակներ

Օրինակ 1

Դիտարկենք հետևյալ դեպքը, երբ $n = 1$ և s -ն այսպիսի տեսք ունի.

```
'1' '0' '0'
'1' '1' '0'
'0' '0' '1'
```

Այս օրինակում մոլորակի մակերևույթը բաղկացած է 3×3 վանդակներից և 2 կղզիներից: Կլիկի `process` ենթածրագրի կանչի միայն 1 փուլ:

0 փուլում գրեյդերը կանչելու է `process` ենթածրագիրը ճիշտ մեկ անգամ.

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

Նկատենք, որ h -ի վանդակների միայն առաջին երեք բիթերն են ցույց տրված:

Ենթածրագիրը պետք է վերադարձնի "0100..." (բաց թողնված բիթերը բոլորը զրոներ են), որտեղ0010 երկուական կոդը հավասար է 2-ի տասական համակարգում: Նկատենք, որ 96 զրոները բաց են թողնված և փոխարինված են ... -ով:

Օրինակ 2

Դիտարկենք հետևյալ դեպքը, երբ $n = 2$ և s -ն այսպիսի տեսք ունի.

```
'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'
```

Այս օրինակում մոլորակի մակերևույթը բաղկացած է 5×5 վանդակներից և 4 կղզիներից: Լինելու է `process` ենթածրագրի կանչերի 2 փուլ:

0 փուլի ժամանակ գրեյդերը `process` ենթածրագիրը կկանչի 9 անգամ.

```
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)
```

Ենթադրենք, որ այդ կանչերը վերադարձնում են այսպիսի արժեքներ. "011", "000", "000", "111", "111", "011", "110", "010", "111" համապատասխանաբար: Այսպիսով, 0 փուլի ավարտից հետո, h -ը կպահի հետևյալ արժեքները.

```
"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 փուլի ժամանակ գրեյդերը `process` ենթածրագիրը կկանչի մեկ անգամ.

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

Վերջում, ենթածրագիրը պետք է վերադարձնի "0010000...." (բաց թողնված բիթերը բոլորը գրոներ են), որտեղ0000100-ը 4-ի երկուական ներկայացումն է: Նկատենք, որ 93 գրոներ բաց են թողնված և նրանց փոխարեն դրված է ...:

Գրեյդերի սմուլշ

Գրեյդերի սմուլշը ներածում է մուտքային տվյալները հետևյալ ձևաչափով.

- տող 1: T
- բլոկ i ($0 \leq i \leq T - 1$). բլոկը ներկայացնում է i -րդ սցենարը:
 - տող 1: n
 - տող $2 + j$ ($0 \leq j \leq 2 \cdot n$): $s[j][0] \ s[j][1] \ \dots \ s[j][2 \cdot n]$

Գրեյդերի սմուլշը արդյունքը տալում է հետևյալ ձևաչափով.

- տող $1 + i$ ($0 \leq i \leq T - 1$). i -րդ սցենարի համար `process` ենթածրագրի վերջին կանչի վերադարձրած արժեքը տասական համակարգում: