Неинтерактивни робот

Младен је као поклон за истакнуте резултате на такмичењима добио једног робота. Како је он изврстан програмер он жели имплементирати вид на том роботу. Сваки пут када роботске очи (камере) направе слику, она се у меморији представља помоћу црних и белих пиксела. Свака слика је представљена као матрица димензија $H \times W$ пиксела, где су редови означени бројевима од 0 до $H - 1$, док су колоне означени бројевима од 0 до $W - 1$. Такође, познато је да се на слици налазе **тачно два** црна пиксела.

Робот слику може процесирати помоћу програма који се састоји од врло једноставних инструкција.

Дате су вредности H, W и позитиван цео број K. **Ваш је задатак да напишете процедуру која ће генерисати низ команди који је способан да за било коју слику димензија $H \times W$ одреди да ли је растојање између два црна пиксела тачно K**. Удаљеност између два пиксела од којих се први налази у врсти r_1 и колони c_1, а други се налази у врсти r_2 и колони c_2 износи $|r_1 - r_2| + |c_1 - c_2|$.

Следи опис рада робота.

Меморију робота замишљамо као довољно велики низ који је индексиран од 0. Сваки елемент тог низа може имати вредност 0 или 1 и ту је вредност након постављања немогуће променити. Слика је у меморији сачувана по врстама, а протеже се од индекса 0 до индекса $H \cdot W - 1$. Прецизније, први врста је сачувана од индекса 0 до индекса $W - 1$, а последња врста сачувана од индекса $(H - 1) \cdot W$ до индекса $H \cdot W - 1$. Ако је пиксел слике у реду i и колони j црн, вредност елемента низа на индексу $i \cdot W + j$ износи 1. У супротном, та вредност износи 0.

Роботски програм је низ **инструкција** које су редом означене целим бројевима почевши од 0. Када се роботски програм покрене, инструкције се извршавају редом , једна по једна. Улаз у сваку инструкцију је један или више елемената из меморије (те вредности називамо **улазним вредностима**), а инструкција враћа једну вредност која износи 0 или 1 (ту вредност називамо **излазном вредношћу**). Излазна вредност i-те инструкције у роботској меморији се чува на индексу $H \cdot W + i$. Улазне вредности у i-ту инструкцију могу бити само меморијске вредности на индексима од 0 до $H \cdot W + i - 1$.

Постоје четири врсте инструкција:

- `NOT`: прима тачно једну улазну вредност. Њена излазна вредност износи 1 ако

улазна вредност износи 0, а иначе излазна вредност износи 0.
- `AND`: прима једну или више улазних вредности. Њена излазна вредност износи 1 ако и само ако **све** њене улазне вредности износе 1.
- `OR`: прима једну или више улазних вредности. Њена излазна вредност износи 1 ако и само ако **бар једна** од њених улазних вредности износи 1.
- `XOR`: прима једну или више улазних вредности. Њена излазна вредност износи 1 ако и само ако је број њених улазних вредности које су једнаке 1 **непаран** .

Излазна вредност последње извршене инструкције роботског програма треба износити 1 ако је удаљеност између црних пиксела са слици једнака K. У противном, излазна вредност последње извршене инструкције треба да буде 0.

Детаљи имплементације

Потребно је имплементирати следећу процедуру:

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

- H, W: димензија слике
- K: позитиван цео број
- Процедура треба да креира роботски програм описан у тексту задатка. Прецизније, за сваку слику одговарајућих димензија, роботски програм треба да одреди да ли је удаљеност између црних пиксела тачно K.

Процедура треба позвати једну или више функција из следећег списка:

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

- Ове функције на крају роботског програма додају `NOT`, `AND`, `OR` или `XOR` инструкцију (редом како је наведено).
- N (за `add_not`): индкес у меморији са којег функција `NOT` чита своју улазну вредност.
- Ns (за `add_and`, `add_or`, `add_xor`): низ који садржи индексе у меморији са којих инструкције `AND`, `OR` или `XOR` читају своје улазне вредности
- Свака функција враћа индекс у меморији у коју та иста инструкција чува своју излазну вредност. Стога узастопни позиви ових функција ће вратити узастопне целе бројеве почевши од $H \cdot W$.

Роботски програм може имати **највише** $10\,000$ **инструкција**. Инструкције смеју **кумулативно примати највише** $1\,000\,000$ улазних вредности. Односно, укупна дужина свих низова Ns у свим позивима функција `add_and`, `add_or` и `add_xor`,

заједно са бројем позива функције `add_not` не сме прелазити $1\,000\,000$.

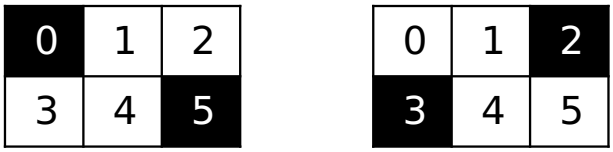
Након позива последње функције, процедура `` `construct_network `` завршава рад. Потом ће се тај добијени роботски програм покренути на одређеном броју слика. Ваше решење пролази неки тест пример ако, за сваку такву слику, излаз последње извршене инструкције износи 1 ако и само ако је удаљеност између црних пиксела на слици једнака K.

Евалуација вашег програма може резултирати једном од следећих порука на енглеском језику.

- `Instruction with no inputs`: празан низ је дат као улазни параметар функција `add_and`, `add_or` или `add_xor`.
- `Invalid index`: Неисправан (можда и негативан) индекс се налази као улаз у `add_and`, `add_or`, `add_xor` или `add_not` funkciju.
- `Too many instructions`: ваша процедура је покушала креирати роботски програм који садржи више од $10\,000$ инструкција.
- `Too many inputs`: инструкције су као улазне податке кумулативно примиле више од $1\,000\,000$ улазних вредности.

Примери

Нека је $H = 2$, $W = 3$, $K = 3$. Постоје само два могуће слике на којима је растојање између црних пиксела тачно 3 (слика испод).



- Случај 1: црни пиксели су на позицијама 0 и 5
- Случај 2: црни пиксели су на позицијама 2 и 3

Један од могућих роботских програма је:

1. `add_and([0, 5])`, додаје инструкцију која враћа 1 ако и само ако се ради о случају 1.Излаз се налази на индексу 6.
2. `add_and([2, 3])`, додаје инструкцију која враћа 1 ако и само ако се ради о случају 2. Излаз се налази на индексу 7.
3. `add_or([6, 7])`, додаје инструкцију која враћа 1 ако и само ако се ради о случају 1 или случају 2.

Ограничења

- $1 \leq H \leq 200$

- $1 \leq W \leq 200$
- $2 \leq H \cdot W$
- $1 \leq K \leq H + W - 2$

Подзадаци

1. (10 поена) $\max(H, W) \leq 3$
2. (11 поена) $\max(H, W) \leq 10$
3. (11 поена) $\max(H, W) \leq 30$
4. (15 поена) $\max(H, W) \leq 100$
5. (12 поена) $\min(H, W) = 1$
6. (8 поена) Пиксел у врсти 0 и колони 0 је црн на свакој слици.
7. (14 поена) $K = 1$
8. (19 поена) Нема додатних огрнаичења.

Грејдер

Грејдер учитава податке у следећем формату:

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

Сви редови осим првог и последњег представљају слике са два црна пиксела. Слика у линији $2 + i$ означавамо бројем i. Један црни пиксел те слике налази се у врсти $r_1[i]$ и колони $c_1[i]$, док се други црни пиксел налази у врсти $r_2[i]$ и колони $c_2[ii]$.

Грејдер прво позива `construct_network(H, W, K)`. Ако позив `construct_network(H, W, K)` нарушава неко од ограничења из текста задатка, грејдер штампа једну од грешака описаних у секцији `Детаљи имплементације` и завршава рад. У супротном, грејдер враћа два излаза.

Грејдер штампа излаз роботског програма у следећем формату:

- линија $1 + i$ ($0 \leq i$): излаз последње инструкције роботског програма за слику i (1 ili 0).

Грејдер штампа следеће податке у датотеку "log.txt" коју ствара у тренутном директоријуму.

- линија $1 + i$ ($0 \leq i$): $m[i][0]$ $m[i][1]$ $\ldots$ $m[i][c-1]$

Низ бројева у $(1 + i)$-вом реду датотеке представља вредности сачуване у роботској меморији након извршавања роботског програма над сликом i.

Прецизније, $m[i][j]$ представља вредност меморијске локације са индексом j. Приметите да дужина једне врсте матрице m (c) одговара броју $H \cdot W$ увећаном за број инструкција роботског програма.