Задатак Digit Sum

Little Digit je, из неког разлога, заинтересована за збир цифара бројева. За сваки ненегативан цео број x, нека је $\mathrm{S}(x)$ збир цифара броја x. На пример

$$\mathrm{S}(123) = 1 + 2 + 3 = 6,$$
$$\mathrm{S}(998877) = 9 + 9 + 8 + 8 + 7 + 7 = 48,$$
$$\mathrm{S}(0) = 0.$$

Њена другарица, Little Square, дала јој је низ $A_1, \dots, A_N$ од N ненегативних целих бројева, као и још један низ $X_1, \dots, X_Q$ од Q ненегативних целих бројева. За свако $i = 1, \dots, Q$, помозите јој да пронађе збир

$$\mathrm{S}(X_i + A_1) + \cdots + \mathrm{S}(X_i + A_N).$$

Улаз

Први ред садржи бројеве N и Q. Наредних $N+Q$ редова садрже бројеве $A_1, \dots, A_N, X_1, \dots, X_Q$, тим редом, сваки у посебном реду.

Излаз

Исписати Q редова. У реду i излаза треба исписати збир

$$\mathrm{S}(X_i + A_1) + \cdots + \mathrm{S}(X_i + A_N).$$

Ограничења

- $1 \leq N, Q \leq 5000$
- $1 \leq A_i, X_i \leq 10^{2000}$

#	Поени	Ограничења
1	19	$X_i, A_i \leq 10^9$
2	18	X_i, A_i се састоје само од цифара 0, 1, 2, 3, 4
3	15	$N, Q \leq 300$ и $X_i, A_i \leq 10^{300}$
4	24	$N, Q \leq 1000$
5	24	Без додатних ограничења

Примери

Улазни подаци	Излазни подаци
5 3	39
10	41
8	42
72	
13	
26	
18	
31	
15	

Објашњење Улазни подаци су

$$\begin{aligned} A_1 &= 10, \\ A_2 &= 8, \\ A_3 &= 72, \\ A_4 &= 13, \\ A_5 &= 26 \end{aligned}$$

и

$$\begin{aligned} X_1 &= 18, \\ X_2 &= 31, \\ X_3 &= 15. \end{aligned}$$

Потребно је израчунати три збира.

Први збир је

$$\begin{aligned} &\mathrm{S}(X_1 + A_1) + \cdots + \mathrm{S}(X_1 + A_5) \\ &= \mathrm{S}(18 + 10) + \mathrm{S}(18 + 8) + \mathrm{S}(18 + 72) + \mathrm{S}(18 + 13) + \mathrm{S}(18 + 26) \\ &= \mathrm{S}(28) + \mathrm{S}(26) + \mathrm{S}(90) + \mathrm{S}(31) + \mathrm{S}(44) \\ &= 2 + 8 + 2 + 6 + 9 + 0 + 3 + 1 + 4 + 4 = 39. \end{aligned}$$

Други збир је

$$\begin{aligned} &\mathrm{S}(X_2 + A_1) + \cdots + \mathrm{S}(X_2 + A_5) \\ &= \mathrm{S}(31 + 10) + \mathrm{S}(31 + 8) + \mathrm{S}(31 + 72) + \mathrm{S}(31 + 13) + \mathrm{S}(31 + 26) \\ &= \mathrm{S}(41) + \mathrm{S}(39) + \mathrm{S}(103) + \mathrm{S}(44) + \mathrm{S}(57) \\ &= 4 + 1 + 3 + 9 + 1 + 0 + 3 + 4 + 4 + 5 + 7 = 41. \end{aligned}$$

Трећи збир је

$$\begin{aligned} &\mathrm{S}(X_3 + A_1) + \cdots + \mathrm{S}(X_3 + A_5) \\ &= \mathrm{S}(15 + 10) + \mathrm{S}(15 + 8) + \mathrm{S}(15 + 72) + \mathrm{S}(15 + 13) + \mathrm{S}(15 + 26) \\ &= \mathrm{S}(25) + \mathrm{S}(23) + \mathrm{S}(87) + \mathrm{S}(28) + \mathrm{S}(41) \\ &= 2 + 5 + 2 + 3 + 8 + 7 + 2 + 8 + 4 + 1 = 42. \end{aligned}$$

Задатак Training

Светско првенство у фудбалу 2026. године се приближава, а Ви, селектор Info(1)Cup репрезентације, припремате интензивне тренинге. Нажалост, хотел који сте резервисали има само један фудбалски терен. Након обиласка, установљавате да терен има дужину N метара и ширину M метара. То би било сасвим у реду када би N и M били у одговарајућим пропорцијама. Проблем је у томе што је $\underline{N \leq 3}$, док је $M \leq 200\,000$.

Сваки од $N \times M$ играча налази се на различитој позицији (i, j) на терену, где $1 \leq i \leq N$ и $1 \leq j \leq M$. Сваки играч припада тачно једном тиму; нека $A_{i,j}$ означава тим играча на позицији (i, j).

Имате два асистента са веома различитим филозофијама: Напад и Одбрана. Они су припремили следећу игру, која се игра *независно за сваки тим.*

1. Одбрана бира једног играча тима који први прима лопту.
2. Затим асистенти наизменично вуку потезе. Напад игра први, затим Одбрана, и тако даље.
3. На потезу, асистент који је на реду говори играчу који тренутно има лопту којем играчу треба да дода лопту. **Тај играч мора бити из истог тима.**
4. Пошто играчи нису у најбољој форми, играч може да дода лопту само саиграчу који се налази или у истом реду или у истој колони терена.
5. Због исцрпљености, ако играч прими лопту више од једном, повредиће се. Асистенти никада неће изабрати ову опцију, јер је она у великој мери против интереса тима.
6. Када тренутни играч нема коме да дода лопту, асистент који је на потезу губи игру.

Оба асистента играју оптимално.

Желите да тестирате како се играчи прилагођавају различитим ситуацијама, па извршавате Q измена. Свака измена има следећи облик:

Играч на позицији (x, y) се распоређује у тим t. Другим речима, постављамо $A_{x,y} = t$.
(Могуће је да је $A_{x,y}$ већ био једнак t.)

Пре било каквих измена, као и након сваке измене, одредите број тимова за које Напад побеђује у игри.

Улазни подаци

У првом реду дати су N и M. У наредних N редова дато је по M елемената. У i-том од ових редова, j-та вредност је $A_{i,j}$. У следећем реду дат је број Q. У наредних Q редова дата су по три цела броја: x, y и t.

Излазни подаци

Исписати $Q + 1$ ред. У првом реду исписати број победничких тимова за Напад пре било каквих измена. У i+1-том реду исписати број победничких тимова за Напад након i-те измене.

Ограничења

- $1 \leq N \leq 3$
- $1 \leq M, Q \leq 200\,000$
- $1 \leq N \times M \leq 200\,000$
- $1 \leq A_{i,j} \leq N \times M$
- За сваку измену важи: $1 \leq x \leq N$, $1 \leq y \leq M$, $1 \leq t \leq N \times M$

#	Поени	Ограничења
1	19	$N = 1, Q \leq 100$
2	20	$N = 1$
3	8	$M = 3$
4	18	$N = 2, Q \leq 100$
5	17	$N = 2$
6	18	Без додатних ограничења

Примери

Улазни подаци	Излазни подаци

`1 6`
`1 2 2 1 2 1`
`3`
`1 1 2`
`1 2 3`
`1 1 3`
	`0`
	`2`
	`1`
	`3`

`2 4`
`1 2 3 1`
`2 2 2 3`
`3`
`1 4 2`
`1 2 1`
`2 4 1`
	`2`
	`0`
	`1`
	`0`

`3 3`
`3 9 1`
`9 3 3`
`3 9 9`
`3`
`2 1 3`
`2 3 9`
`3 3 1`
	`1`
	`0`
	`2`
	`2`

Објашњења

Погледајмо први пример. Пре извршавања било каквих измена, Одбрана може да победи у игри за оба тима. На пример, анализирајмо тим 1:

Претпоставимо да Одбрана да лопту играчу који се налази на позицији $(1, 4)$. Затим, Напад каже играчу на позицији $(1, 4)$ да дода лопту играчу на позицији $(1, 1)$. Сада је једина опција за Одбрану да играч са позиције $(1, 1)$ дода лопту играчу на позицији $(1, 6)$. Пошто у тиму 1 не постоје други играчи, играч на позицији $(1, 6)$ не може да дода лопту никоме другом. Пошто је тада на потезу Напад, он губи.

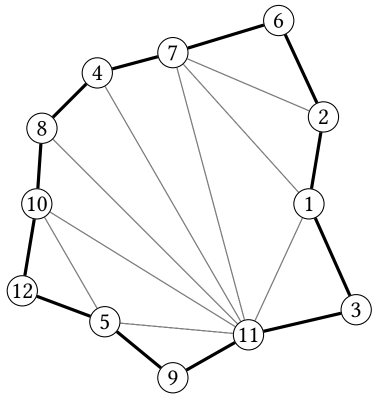
Након прве измене, терен изгледа на следећи начин:

```
2 2 2 1 2 1
```

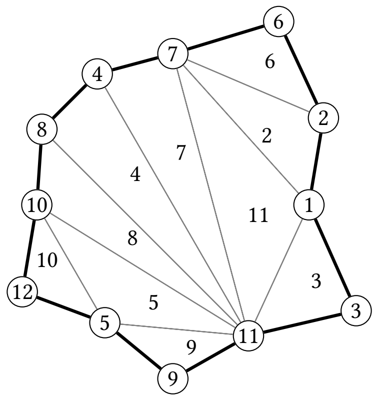
За тим 1, ако Одбрана да лопту играчу на позицији $(1,4)$, Напад ће му рећи да пошаље лопту на позицију $(1,6)$. Сада не постоји ниједан играч коме се може додати лопта. Стога, Напад побеђује у игри за тим 1.

Задатак Triangulation

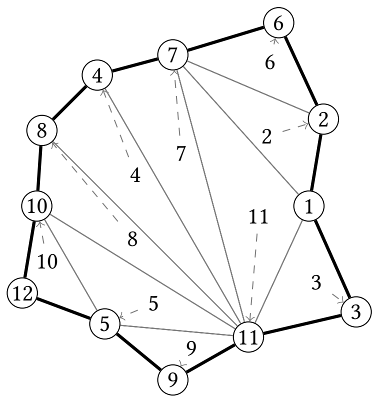
Little Polygon је добио поклон од свог пријатеља Little Circle. У поклону се налазио цртеж многоугла од N темена, где је свако теме било означено целим бројем од 1 до N, и његова триангулација. Пример цртежа:



Такође, у поклону се налазила и порука у којој се од Little Polygon тражи да у сваком од троуглова напише по један цео број од 2 до $N-1$, тако да сваки број буде написан тачно једном и тако да број i буде написан унутар троугла чије је једно од темена означено са i. На пример:



Имајте у виду да је сваки број од 2 до $N-1$ записан тачно једном и да је број i записан унутар троугла чије је једно од темена означено са i:



Помозите Little Polygon да нађе начин да запише ове бројеве у троуглове, као и да преброји колико постоји начина да се то уради.

Улаз

Сваки улазни фајл садржи више тест примера. У првој линији улаза налази се број T, број тест примера. Затим, у првој линији сваког тест примера налази се број N, број страница датог многоугла. Друга линија садржи индексе темена многоугла, у смеру супротном од смера казаљке на сату. Наредне $N-2$ линије садрже по три броја који представљају темена по једног троугла дате триангулације.

Излаз

За сваки тест пример исписати по две линије. У првој линији исписати један валидан начин да се запишу бројеви у троуглове (исписати бројеве у истом редоследу у ком су троуглови дати у улазу). У другој линији исписати колико постоји валидних начина да се ово уради, по модулу $10^9 + 7$.

Ограничења

- Ако је само прва линија сваког тест примера у излазном фајлу тачна, освојићете 50% поена за тај фајл. Ако је само друга линија сваког тест примера у излазном фајлу тачна, такође ћете освојити 50% поена за тај фајл. Ако је цео излазни фајл тачан, освојићете 100% поена за тај фајл.
- **Напомена!** Исписати по две линије за сваки тест пример, прва садржи $N-2$ целих бројева, а друга садржи један цео број, тако да је сваки од њих могуће представити као тип променљиве `int`, чак и ако је једна од линија погрешна. У супротном, тај излазни фајл ће бити оцењен као `Invalid output format`.
- $1 \leq T \leq 50\,000$.
- $3 \leq N \leq 50\,000$.
- Сума свих N у свим тест примерима унутар једног фајла је највише 50 000
- У сваком тест примеру троуглови који су дати формирају валидну триангулацију.

#	Поени	Ограничења
1	8	$T \leq 10$ и $N \leq 8$
2	11	$T \leq 10$ и $N \leq 100$
3	23	$T \leq 10$ и $N \leq 1000$
4	17	1 и N су суседни у многоуглу
5	14	сви троуглови садрже теме 1
6	9	сви троуглови садрже теме 2
7	18	без додатних ограничења

Примери

Улазни подаци	Излазни подаци	Објашњење

`1`
`12`
`1 2 6 7 4 8 10 12 5 9 11 3`
`11 5 9`
`1 7 11`
`10 5 12`
`1 2 7`
`6 7 2`
`3 11 1`
`11 10 8`
`11 10 5`
`4 8 11`
`7 4 11`
	`9 11 10 2 6 3 8 5 4 7`
	`9`
		Ово је пример који је нацртан горе, у поставци задатка.

Задатак Grow Gravity

Little Newton воли гравитацију и игра се магичним обликом направљеним од 1×1 квадрата, који лебде у ваздуху.

На почетку, овај облик се састоји од само једног таквог квадрата. Над овим магичним обликом дешавају се две ствари (тим редом):

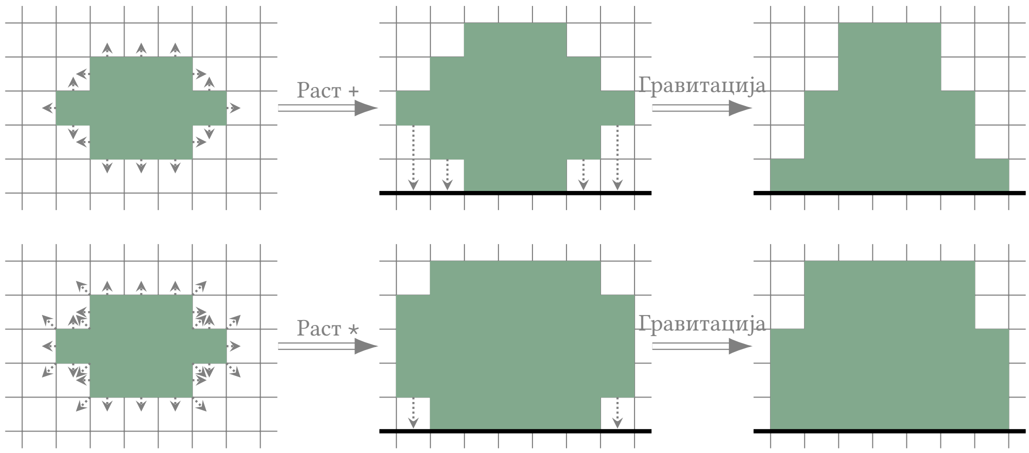
1. Фаза раста: Око сваког квадрата у облику појављују се нови квадрати. Фаза раста може бити једног од два типа:

   Тип +. Нови квадрати се појављују само у 4 правца (север, југ, исток и запад).

   Тип ⋆. Нови квадрати се појављују у 8 праваца (север, југ, исток, запад, североисток, северозапад, југоисток и југозапад).

2. Фаза гравитације: У овој фази, Little Newton поставља плочу испод облика и пушта све квадрате да падну на њу. **Напомена:** Плоча се одмах након ове операције уклања!

Као пример, ефекти оба типа фаза раста, након којих следи фаза гравитације, приказани су на сликама испод.



Little Newton има низ $v_1, \ldots, v_N$ од N типова фаза раста и жели да изврши Q операција следећих типова:

- update(i): Ако је $v_i = +$, поставити га на $\star$; и обрнуто, ако је $v_i = \star$, поставити га на $+$.
- query(ℓ, r): Претпоставимо да Little Newton почиње са обликом који се састоји од само једног квадрата величине 1×1 и да за $i = \ell, \ldots, r$, редом, примењује фазу раста типа v_i, након које следи фаза гравитације. Колико ће квадрата величине 1×1 облик имати на крају?

Помозите Little Newton-у да одговори на све упите.

Улазни подаци

Први ред улаза садржи број N. Други ред садржи N симбола из скупа $\{+, \star\}$, без размака, који представљају почетне вредности $v_1, \ldots, v_N$. Трећи ред улаза садржи број Q. Затим, сваки од наредних Q редова садржи једну од следећих операција:

- или два цела броја 1 i, што представља операцију update(i),
- или три цела броја 2 ℓ r, што представља операцију query(ℓ, r).

Излазни подаци

За сваку операцију типа query(ℓ, r) потребно је исписати одговор, сваки у посебном реду.

Ограничења

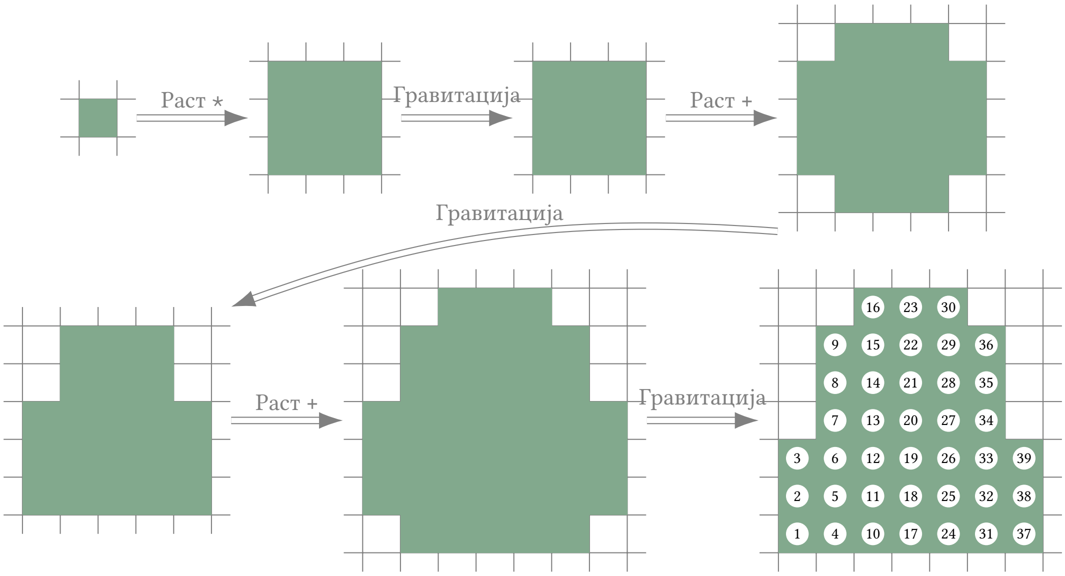
- $1 \leq N \leq 200\,000$.
- $1 \leq Q \leq 200\,000$.
- $v_i \in \{+, \star\}$ за $1 \leq i \leq N$.
- $t \in \{1, 2\}$, $1 \leq i \leq N$ и $1 \leq \ell \leq r \leq N$ за сваку операцију.

#	Поени	Ограничења
1	5	$v_i = \star$ за свако $1 \leq i \leq N$ и операције су само типа 2.
2	10	$1 \leq N \leq 40$ и $1 \leq Q \leq 40$.
3	20	$1 \leq N \leq 300$ и $1 \leq Q \leq 300$.
4	30	$v_i = +$ за свако $1 \leq i \leq N$ и операције су само типа 2.
5	35	Без додатних ограничења.

Улазни подаци	Излазни подаци
4	39
+*++	49
9	31
2 2 4	
1 3	
1 4	
2 2 4	
1 1	
1 2	
1 3	
1 4	
2 2 4	

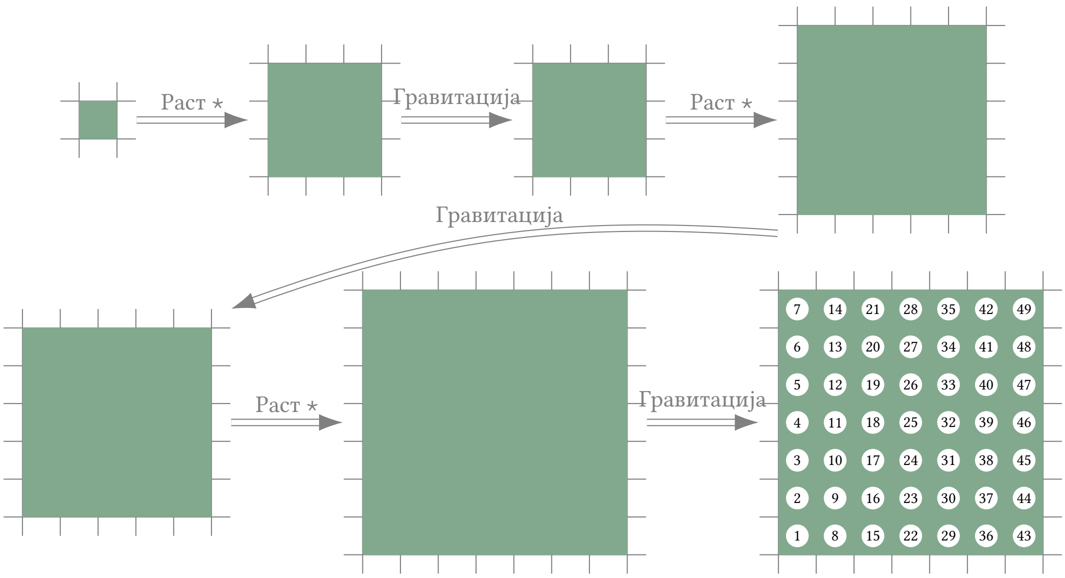
Објашњења

Први упит. У првом упиту, низ операција је $\star$++. Облик тада има следеће конфигурације.



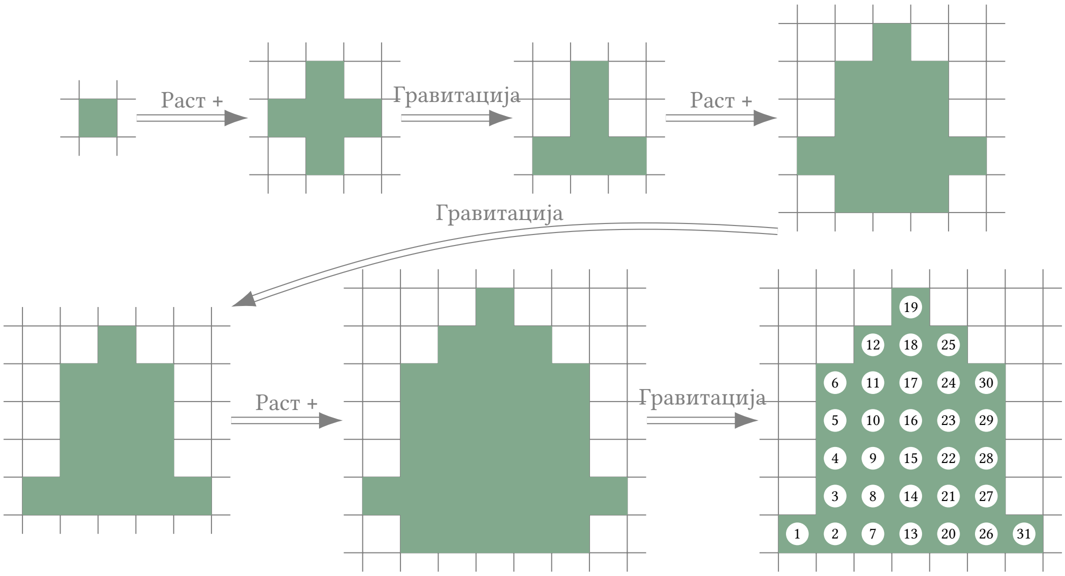
Зато је одговор 39.

Други упит. У другом упиту, низ операција је ⋆⋆⋆. Облик тада има следеће конфигурације.



Дакле, одговор је 49.

Трећи упит. У последњем упиту, низ операција је +++. Облик тада има следеће конфигурације.



Стога је одговор 31.

Задатак Hearts

Little Green Heart је поклонио Little Purple Heart за Дан заљубљених два низа A и B дужине N такве да они заједно садрже све бројеве од 1 до $2 \times N$ тачно једном.

Little Purple Heart извршава следећу операцију:

1. Бира два индекса l и r, $1 \leq l \leq r \leq N$.
2. Затим креира низ C дужине $k = r - l + 1$, такав да за свако i, $0 \leq i < k$, бира или $C_i = A_{l+i}$ или $C_i = B_{l+i}$.
3. Затим рачуна вредност $\text{val}(C)$, која је једнака броју индекса i, $0 \leq i < k$, таквих да не постоји индекс j за који важи $0 \leq j < i$ и $C_j > C_i$.

Little Green Heart, као прави програмер, жели да импресионира Little Purple Heart тако што ће израчунати суму $\text{val}(C)$ за све низове C које она може да направи. Пошто тај број може да буде прилично велики, он ће израчунати његов остатак при дељењу са $10^9 + 7$. Додатно, он жели да уради то за Q различитих операција.

Формално, за сваки од датих Q упита облика (l, r), $1 \leq l \leq r \leq N$, он дефинише S као скуп свих низова C које Little Purple Heart може да направи. Затим, за сваки упит, он жели да израчуна

$$P(l, r) = \sum_{C \in S} \text{val}(C) \mod 10^9 + 7$$

Ваш задатак је да помогнете Little Green Heart да импресионира Little Purple Heart тако што ћете израчунати решење за сваки упит.

Улаз

У првој линији улаза налази се број N. Друга линија садржи N бројева $A_1, A_2, \ldots, A_N$ који представљају елементе низа A. Трећа линија садржи N бројева $B_1, B_2, \ldots, B_N$ који представљају елементе низа B. Ова два низа заједно садрже сваки број од 1 до $2 \times N$ тачно једном.

Четврта линија садржи број Q који представља број упита. Свака од наредних Q линија садржи по 2 броја који представљају упит облика l r.

Излаз

Излаз садржи Q линија које представљају одговоре на упите $P(l, r)$, сваки у посебном реду, израчунате по модулу $10^9 + 7$.

Ограничења

- $1 \leq N \leq 70\,000$.
- $1 \leq Q \leq 70\,000$.

#	Поени	Ограничења
1	5	$1 \leq N \leq 17$, $1 \leq Q \leq 100$
2	16	$1 \leq N, Q \leq 300$
3	9	$B_i = A_i + 1$ за свако i, $1 \leq i \leq N$
4	17	$1 \leq N, Q \leq 10\,000$
5	11	$1 \leq Q \leq 200$
6	14	$1 \leq N \leq 50\,000$, $1 \leq Q \leq 10\,000$
7	10	$1 \leq N, Q \leq 50\,000$
8	18	Без додатних ограничења

Примери

Улазни подаци	Излазни подаци

`6`
`2 10 3 4 5 6`
`11 1 9 8 7 12`
`2`
`3 6`
`1 2`
	`37`
	`5`

`12`
`11 10 7 23 1 18 22 16 8 14 20 6`
`3 4 9 13 19 5 24 12 15 21 17 2`
`1`
`7 11`
	`32`

Објашњења примера

Први пример. Постоје два упита.

У првом упиту, разматрамо поднизове A_3, A_4, A_5, A_6 и B_3, B_4, B_5, B_6 помоћу којих конструишемо низ C. Затим разматрамо све могуће опције за низ C и суму свих $\text{val}(C)$. На пример, за $C = [3, 8, 7, 12]$, индекси $i = 0, 1, 3$ имају особину да не постоји ниједан индекс $0 \leq j < i$ такав да $C_j > C_i$. Због тога је $\text{val}(C) = 3$.

Сумирањем $\text{val}(C)$ за све низове C који се могу конструисати, добијамо $P(3, 6) = 37$.

За други упит, можемо конструисати четири низа C:

$P(1, 2) = \text{val}([2, 10]) + \text{val}([2, 1]) + \text{val}([11, 10]) + \text{val}([11, 1]) = 2 + 1 + 1 + 1 = 5$.

Други пример. Постоји један упит. Израчунавамо $P(7, 11) = 32$.

Задатак Trees

Little Tree је заинтересована за стабла разних величина. Дато јој је једно укорењено бинарно стабло T (тј. такво да сваки чвор има највише 2 детета), *не нужно балансирано (уравнотежено)*, чији је корен означен бројем 1, а чворови означени бројевима $1, \ldots, 2^N$. Ознака сваког чвора је увек строго мања од ознака његове деце.

Скуп чворова S назива се *валидним* ако ниједан чвор из S није предак било ког другог чвора из S. Вредност скупа S, означена са $\mathrm{val}(S)$, једнака је броју чворова у стаблу који имају *бар један* чвор из S за претка (сматрамо да је чвор сам себи предак). Другим речима, то је збир величина подстабала укорењених у чворовима из S.

Њен непријатељ, Little Cactus, изазива је следећим задатком: дат је број N и стабло T, потребно је одредити што више *валидних*, *непразних* скупова $S_1, \ldots, S_K$ таквих да важи

$$|S_1| + \cdots + |S_K| \leq N \times 2^{N-1} + 1$$

и да је $\mathrm{val}(S_i) \neq \mathrm{val}(S_j)$ за свака два $i \neq j$.
Напоменимо да смо са $|S_i|$ означили број елемената скупа S_i.

Улазни подаци

Прва линија улазног фајла садржи цео број N. Друга линија садржи целе бројеве $p_2, \ldots, p_{2^N}$, раздвојене размацима, где p_i означава родитеља чвора i.

Излазни подаци

Исписати број K, који представља број скупова које сте успели да одредите, а затим сваки од K скупова у посебном реду. За испис једног скупа, најпре исписати његову величину, а затим његове елементе, раздвојене размацима.

Ограничења

- $N \leq 15$.
- Ако испишете неисправно решење (тј. користите превише чворова или два скупа имају исту вредност), добићете 0 поена за тај тест пример. У супротном, део укупног броја поена који добијате за одређени тест једнак је $\left(\frac{K}{2^N}\right)^{1,5}$.

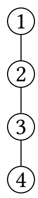
#	Поени	Ограничења
1	7	$p_i = i - 1$ за $i = 2, \ldots, 2^N$
2	12	$N = 4$
3	13	$N = 6$
4	21	$N = 9$
5	26	$N = 11$
6	21	Без додатних ограничења

Примери

Улазни подаци	Излазни подаци
`2`	`4`
`1 2 3`	`1 4`
	`1 3`
	`1 2`
	`1 1`
`2`	`4`
`1 2 2`	`1 4`
	`2 3 4`
	`1 2`
	`1 1`

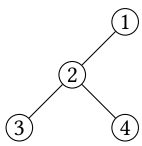
Објашњења

Први пример. Стабло изгледа као на слици:



Видимо да скупови $S_1 = \{4\}$, $S_2 = \{3\}$, $S_3 = \{2\}$, $S_4 = \{1\}$ имају међусобно различите вредности (редом 1, 2, 3, 4) и да је њихова укупна величина $4 \leq 2 \times 2^{2-1} + 1 = 5$. Према томе, овај излаз је исправан.

Други пример. Стабло изгледа као на слици:



Видимо да скупови $S_1 = \{4\}$, $S_2 = \{3, 4\}$, $S_3 = \{2\}$, $S_4 = \{1\}$ имају међусобно различите вредности (редом 1, 2, 3, 4) и да је њихова укупна величина $5 \leq 2 \times 2^{2-1} + 1 = 5$. Према томе, овај излаз је исправан.