

Задатак Nines

Улаз `stdin`
Излаз `stdout`

Дат је цео број $N > 0$ и број K . Прво, за сваки природан број x дефинишимо

$$P_K(x) = \max((\text{број појављивања цифре } 9 \text{ у } x) - K, 0).$$

Затим дефинишимо *вредност* целог броја x , у ознаци $\text{value}_K(x)$, са

$$\text{value}_K(x) = x + 10^{P_K(x)}.$$

Ако су дати N и K , која је највећа могућа вредност неког ненегативног целог броја мањег од N ? Другим речима, израчунати

$$\text{answer}(N, K) = \max_{0 \leq x < N} \text{value}_K(x).$$

Улазни подаци

Прва линија улаза садржи број тест случајева T . Наредних $2T$ редова описују тест случајеве. Прва линија тест случаја садржи два броја: D (представља број цифара броја N) и K . Друга линија тест случаја садржи цео број N .

Излазни подаци

Исписати T редова који представљају одговоре за T вредности N и K које су дате.

Додатна ограничења

- Нека је $\sum D$ збир свих D у једном тест примеру.
- $1 \leq T \leq 100\,000$.
- $1 \leq \sum D \leq 1\,000\,000$.
- $-1 \leq K \leq D$.

#	Поени	Додатна ограничења
1	7	$T \leq 1\,000, D \leq 4$.
2	3	$K = -1, D \leq 9$.
3	6	$K = 0, D \leq 9$.
4	6	$D \leq 9$.
5	4	$K = D$
6	26	$T \leq 200, D \leq 200$.
7	48	Без додатних ограничења.

Улаз	Израз
2	199
3 0	1097999999999999999
100	
18 0	
998877665544332211	

Објашњење примера

За први тест случај оптимално је $\text{value}_0(99) = 199$.

У другом тест случају је оптимално

$$\text{value}_0(997\,999\,999\,999\,999\,999) = 1097999999999999999.$$

Задатак Cards

Улаз `stdin`
Излаз `stdout`

LITTLE CS игра игру са N карата, које су нумерисане бројевима од 1 до N , као и са столом на коме нема карата. Свака карта на себи има написан позитиван цео број. Бројеви су написани тако да не постоје две карте са истим бројем. Игра се игра на следећи начин:

1. LITTLE CS послаже карте у шпил у произвољном редоследу. Нека је $p_1, p_2, \dots, p_N$ редослед карата у шпиљу након што их је LITTLE CS послагао, где је p_i број написан на i -тој карти.
2. Све док шпил није празан LITTLE CS узима прву карту из шпиља и ради следеће:
 - Ако је карта *доминантна*, он је ставља на сто. Карта је доминантна ако на столу не постоји ниједна карта, или је вредност на тренутној карти барем два пута већа него свака вредност на картама на столу.
 - Ако карта није доминантна, он је баца и не може више да је користи.
3. Када шпил карата постане празан, игра се завршава. Нека је S сума вредности карата на столу. Онда је S резултат игре.

LITTLE CS воли да се такмичи и жели да му крајњи резултат буде што је већи могући. Занима га највећи могући резултат игре. Другим речима, занима га која је највећа вредност броја S коју може да добије ако оптимално изабере вредности $p_1, p_2, \dots, p_N$. LITTLE CS-а такође занима број различитих редоследа карата $p_1, p_2, \dots, p_N$ који доводи до највеће могуће вредности броја S .

LITTLE CS-у је ово лак задатак, тако да је одлучио да га отежа и да Вам га зада. Он је изабрао N карата са вредностима $a_1, a_2, \dots, a_N$ на њима, као и Q парова целих бројева (l, r) , таквих да је $1 \leq l \leq r \leq N$. За сваки од парова Вас пита да нађете две вредности које га занимају ако би играо игру само са картама које имају вредности $a_l, a_{l+1}, \dots, a_r$.

Опис улаза

У првој линији улаза се налазе два броја, N , број карата, и Q , број упита на које треба одговорити. Друга линија садржи бројеве $a_1, a_2, \dots, a_N$, вредности написане на LITTLE CS-овим картама. Свака од следећих Q линија садржи два броја, i -та линија садржи i -ти пар (l_i, r_i) који је LITTLE CS изабрао за Вас.

Опис излаза

Свака од Q линија стандардног излаза треба да садржи два броја раздвојена размаком, одговоре на LITTLE CS-ова питања. Формалније, i -та линија мора имати следеће две вредности следећим редом:

- Максимална вредност S коју LITTLE CS може да добије играјући само са картама нумерисаним бројевима од l_i до r_i .
- Број низова $p_1, p_2, \dots, p_{r_i-l_i+1}$ који максимизују вредност S . Пошто овај број може да буде веома велик, треба да испишете његов остатак при дељењу са $10^9 + 7$.

Детаљи имплементације

У свом коду такође можете да користите следећу функцију.

```
int product(int a, int b);
```

Функција враћа у константном времену вредност $a \times (a+1) \times \dots \times (b-1) \times b \bmod 10^9 + 7$, тј. производ свих целих бројева од a до b , по модулу $10^9 + 7$. Пазите на то да за вредности параметара a и b мора да важи $1 \leq a \leq b \leq 200\,000$.

Запамтите да учитате (енгл. `include`) хедер `cards.h`, користећи команду `#include "cards.h"`! Пазите да у овом задатку свакако треба да имплементирате `main` функцију.

Ограничења

- $1 \leq N \leq 200\,000$
- $1 \leq Q \leq 50\,000$
- $1 \leq a_i \leq 500\,000$, за свако $1 \leq i \leq N$, све вредности a_i су међусобно различите
- $1 \leq l_i \leq r_i \leq N$, за свако $1 \leq i \leq Q$
- 50% поена за сваки подзадатак добијате ако је највећа вредност S тачно израчуната за све упите, док других 50% добијате ако је број таквих низова тачан за све упите. Пазите да како бисте добили поене за налажење вредности S свакако треба да испишете број низова за сваки упит, чак иако он није тачан.

#	Бодови	Ограничења
1	8	$Q \leq 300, r_i - l_i \leq 7$ за свако $1 \leq i \leq Q$
2	15	$N, Q \leq 300$
3	9	$N, Q \leq 5\,000$
4	10	$a_1 < a_2 < \dots < a_{N-1} < a_N$
5	19	$N \leq 5000$
6	21	$N, Q \leq 40\,000$
7	18	Нема додатних ограничења

Улаз	Излаз
6 2 4 1 7 12 5 21 1 4 4 6	17 2 26 1
5 4 1 3 7 9 15 1 4 2 5 4 5 3 4	13 1 25 2 15 1 9 1
10 6 115 4 319 19 28 1032 516 140 9 710 1 3 2 7 4 10 5 7 1 9 6 10	438 1 1580 8 1725 10 1576 1 1729 48 1697 2

Објашњење примера

За сваки упит, бројеви на картама које буду постављене на сто су подебљани (енгл. bold).

Први пример У првом упиту, вредност S је максимална за следеће низове:

- 1, **4**, 7, **12**
- 1**, **4**, **12**, 7

У другом упиту, вредност S је максимална за следеће низове:

- 5**, **21**, 12

Задатак Ski Circuit

Улаз `stdin`
Излаз `stdout`

Након напуштања Лос Анђелеса, LITTLE AD је стигао у INFO(1)CUP SKI RESORT, највеће скијалиште у INFO(1)CUP KINGDOM. Скијалиште се састоји од $N+1$ тачака нумерисаних бројевима од 0 до N . Висина тачке i је H_i , и $H_0 = 0$. Не постоје две тачке које имају исту висину.

За сваке две тачке i и j за које важи $H_i < H_j$, постоји једна жичара која одводи скијаше од тачке i до тачке j и једна низбрдица која одводи скијаше од тачке j до тачке i . Спуштање низ низбрдицу од тачке j до тачке i траје тачно $H_j - H_i$ секунди. Имајте у виду да када се скијаши спуштају низбрдицом од тачке j до тачке i , они **неће** морати да посете све тачке p за које важи $H_i < H_p < H_j$. Додатно, када се крећу жичаром од тачке i до тачке j , скијаши се не могу зауставити ни у једној тачки $p \neq j$.

Скијашу је потребно U_i секунди да се попне на жичару i (без обзира на то куда та жичара води) и C_i секунди да сиђе са жичаре i (без обзира на то одакле та жичара креће). $U_0 = C_0 = 0$.

LITTLE AD жели да направи K -секундну скијашку туру. Прецизније, он жели да пође из тачке 0, посети свих осталих N тачака тачно једном, и затим да се врати у тачку 0, и да при томе **бар** K секунди проведе скијајући по низбрдицама.

LITTLE AD мрзи да стоји у једној тачки превише дуго (као и сваки страствени скијаш). Према томе, нека је W_i укупно време које LITTLE AD проведе у тачки i , и нека је M максимална вредност свих W_i . Познато је да важи $W_0 = 0$ и, за свако $1 \leq i \leq N$:

- $W_i = 0$, ако LITTLE AD стиже у тачку i низбрдицом и напушта тачку i настављајући да се креће низбрдицом,
- $W_i = C_i$, ако LITTLE AD стиже у тачку i жичаром и напушта тачку i настављајући свој пут низбрдицом,
- $W_i = U_i$, ако LITTLE AD стиже у тачку i низбрдицом и напушта тачку i жичаром,
- $W_i = U_i + C_i$ ако LITTLE AD стиже у тачку i жичаром и напушта тачку i жичаром.

Сада се LITTLE AD пита: колико најмање времена треба да проведе чекајући у некој тачки тако да укупно проведе бар K секунди скијајући низбрдицама? Другим речима, која је минимална вредност M која се може постићи тако да он проведе бар K секунди крећући се низбрдицама?

LITTLE AD-у је потребна ваша помоћ! За T датих сценарија, потребно је да нађете минималну вредност за M и тако помогнете LITTLE AD-у да ужива у свом заслуженом скијашком одмору!

Улаз

У првој линији стандардног улаза налази се број T , који представља број сценарија које ваш програм треба да реши. За сваки од T сценарија, у једној линији уносе се бројеви N и K . Затим, i -та од наредних N линија садржи три вредности H_i , U_i и C_i . Имајте у виду да се са улаза не учитавају вредности H_0 , U_0 и C_0 , пошто је познато да је $H_0 = U_0 = C_0 = 0$.

Излаз

На стандардном излазу исписати тачно T линија, при чему је у i -тој линији број који представља одговор за i -ти сценарио.

Додатна ограничења

- Гарантује се да LITTLE AD може скијати бар K секунди.
- $1 \leq T \leq 200$.
- Нека $\sum N$ представља суму вредности N за све сценарије.
- $1 \leq \sum N \leq 200\,000$.
- $1 \leq K \leq 10^{12}$
- $1 \leq H_i, U_i, C_i \leq 10^6$, за свако $1 \leq i \leq N$

#	Поени	Додатна ограничења
1	6	$1 \leq \sum N \leq 10$
2	9	$1 \leq \sum N \leq 17$
3	19	$1 \leq \sum N \leq 300$
4	22	$1 \leq \sum N \leq 2\,000$
5	44	Без додатних ограничења.

Улаз	Излаз
2	2
3 5	8
1 8 6	
5 3 2	
2 6 8	
3 6	
1 8 6	
5 3 2	
2 6 8	

Објашњења примера

Први сценарио. LITTLE AD се пење на жичару у тачки 0, затим силази са жичаре у тачки 2 и онда скија дуж низбрдице до тачке 3. Одатле наставља да скија до тачке 1, па затим и до тачке 0.

$$W_0 = W_1 = W_3 = 0; W_2 = 2$$

Други сценарио. LITTLE AD се пење на жичару у тачки 0, којом иде до тачке 2, где силази са ње. Затим скија низбрдо до тачке 1. У тачки 1 се пење на жичару и њом иде до тачке 3, где напушта жичару и скија низбрдо до тачке 0.

$$W_0 = 0; W_1 = W_3 = 8; W_2 = 2$$

Задатак Modulosum

C++ header modulosum.h

Следећи задатак се преноси кроз многе генерације у INFO(1)CUP краљевству — можете ли га решити?

Дат Вам је низ $A_0, \dots, A_{N-1}$ који садржи N позитивних целих бројева. Такође Вам је дато Q упита представљених као тројке (L_i, R_i, M_i) за $i = 0, \dots, Q - 1$. За сваки упит треба да израчунате

$$(A_{L_i} \bmod M_i) + \dots + (A_{R_i} \bmod M_i).$$

На пример, ако је $L = 2, R = 3, M = 5, A_2 = 3, A_3 = 4$, онда је одговор $(3 \bmod 5) + (4 \bmod 5) = 7$.

Детаљи имплементације

Треба да имплементирате следећу функцију.

```
std::vector<long long> solve(  
    int N, int Q,  
    std::vector<int> A,  
    std::vector<int> L,  
    std::vector<int> R,  
    std::vector<int> M  
);
```

Функција треба да врати одговор да задатих Q упита. Биће позвана *само једном* по покретању комисијског грејдера. Пазите да је низ A индексан од 0 до $N - 1$, а низови L, R, M су индексани од 0 до $Q - 1$.

Не заборавите да укључите (енгл. include) хедер modulosum.h! Такође, припазите на то да *не треба* да имплементирате main функцију, већ само solve.

Објашњење грејдера датог у прилогу (енгл. sample grader behaviour)

Грејдер ће да учита вредности N, Q , затим низ A , а после тога и Q тројки (L_i, R_i, M_i) . Након тога ће да позове функцију $\text{solve}(N, Q, A, L, R, M)$, и исписаће враћене вредности на стандардни излаз, једну вредност по линији. Следећи фајлови са улазом/излазом раде за дати грејдер.

Додатна ограничења

- $1 \leq N \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $0 \leq L_i \leq R_i < N$
- $1 \leq M_i \leq 1\,000\,000\,000$
- $1 \leq A_i \leq 300\,000$.

#	Поени	Додатна ограничења
1	8	$1 \leq N, Q \leq 1\,000$
2	5	$M_i = M_j, 1 \leq i, j \leq Q$
3	9	$1 \leq N \cdot M_i \leq 2\,000\,000$
4	11	$1 \leq N \cdot M_i \leq 200\,000\,000$
5	25	$10\,000 \leq M_i$
6	42	Нема додатних ограничења.

Примери

Улаз	Израз
5 4	2
5 1 4 2 3	4
0 2 4	3
1 4 3	15
2 3 3	
0 4 10	

Објашњење

У првом упиту, $(5 \bmod 4) + (1 \bmod 4) + (4 \bmod 4) = 1 + 1 + 0 = 2$.

У другом упиту, $(1 \bmod 3) + (4 \bmod 3) + (2 \bmod 3) + (3 \bmod 3) = 1 + 1 + 2 + 0 = 4$.

У трећем упиту, $(4 \bmod 3) + (2 \bmod 3) = 1 + 2 = 3$.

У четвртом упиту,

$(5 \bmod 10) + (1 \bmod 10) + (4 \bmod 10) + (2 \bmod 10) + (3 \bmod 10) = 5 + 1 + 4 + 2 + 3 = 15$.

Задатак Divisors

Улаз `stdin`
Излаз `stdout`

Дат је број N . Ваш задатак је да пронађете број низова $a_1, \dots, a_k$ таквих да је $a_1 \mid a_2 \mid \dots \mid a_k$ (тако да, a_1 дели a_2 , a_2 дели a_3 , и тако редом), и $N = a_1 + \dots + a_k$.

На пример, ако је $N = 6$, низови који задовољавају услове задатка су

1. 6
2. 1, 5
3. 1, 1, 4
4. 1, 1, 1, 3
5. 1, 1, 1, 1, 2
6. 1, 1, 1, 1, 1, 1
7. 1, 1, 2, 2
8. 2, 4
9. 2, 2, 2
10. 3, 3

Дакле одговор би био 10.

Улаз

Прва линија стандардног улаза садржи број тестова T . Наредних T линија садрже вредности броја N за које треба да пронађете одговоре.

Излаз

У сваком од T редова стандардног излаза исписати по један број који представља одговор за одговарајуће N које је дато. Пошто ове вредности могу бити велике, треба исписати њихов остатак при дељењу са $10^9 + 7$.

Додатна ограничења

- $1 \leq T \leq 100\,000$.
- $1 \leq N \leq 500\,000$.

#	Поени	Додатна ограничења
1	12	$N \leq 20$
2	21	$N \leq 100$
3	33	$N \leq 1000$
4	34	Без додатних ограничења.

Примери

Улаз	Израз
3	10
6	26
10	475702494
500000	

Објашњење

Први тест пример је објашњен у поставци задатка. За $N = 10$ низови који задовољавају услове задатка су

1. 10
2. 1, 9
3. 1, 1, 8
4. 1, 1, 1, 7
5. 1, 1, 1, 1, 6
6. 1, 1, 1, 1, 1, 5
7. 1, 1, 1, 1, 1, 1, 4
8. 1, 1, 1, 1, 1, 1, 1, 3
9. 1, 1, 1, 1, 1, 1, 1, 1, 2
10. 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
11. 1, 1, 1, 1, 1, 1, 2, 2
12. 1, 1, 1, 1, 2, 4
13. 1, 1, 1, 1, 2, 2, 2
14. 1, 1, 1, 1, 3, 3
15. 1, 1, 2, 6
16. 1, 1, 2, 2, 4
17. 1, 1, 2, 2, 2, 2
18. 1, 1, 4, 4
19. 1, 3, 6
20. 1, 3, 3, 3
21. 2, 8
22. 2, 2, 6
23. 2, 2, 2, 4
24. 2, 2, 2, 2, 2
25. 2, 4, 4
26. 5, 5

Дакле, одговор је 26.

Задатак Radio

Улаз `stdin`
Излаз `stdout`

Дата је мрежа од N радија повезаних са $N-1$ комуникационих канала. Мрежа радија и комуникационих канала се моделује као стабло. Сваки радио може бити у једном од три стања: нема сигнала (NS), има сигнала (S), или деактивиран (D). Једном када радио дође у стање S или D, он остаје у том стању трајно, а радио у стању NS може да мења стање.

Сваког дана се следећи догађаји дешавају истовремено: сваки радио у стању S ће покушати да се емитује на сваком радију са којим је директно повезан комуникационим каналом. Посматрајмо било који радио r који је у стању NS, и претпоставимо да је примио t сигнала тог дана. Ако је $t = 0$, онда радио r остаје у стању NS наредног дана. Ако је $t = 1$, онда радио r прелази у стање S трајно (почевши од наредног дана). Коначно, ако је $t > 1$, онда радио r прелази у стање D трајно (почевши од наредног дана).

Имајте на уму да ће, без обзира на тренутно стање мреже, постојати дан од ког ниједан радио више никада неће променити стање. Нека се ово стање мреже зове *стабилно стање*.

LITTLE STEVE треба да одговори на Q упита облика: „Претпоставимо да је M радија $x_1, \dots, x_M$ у почетном тренутку у стању S, а да су сви остали у стању NS. Колико радија ће бити у стању S када мрежа достигне своје стабилно стање?” Можете ли да одговорите на LITTLE STEVE-ове упите?

Улазни подаци

Прва линија стандардног улаза садржи број N . Наредних $N-1$ редова садрже парове (x, y) , који означавају комуникационе канале између радија. Наредни ред садржи број Q . Наредних Q редова представљају по један упит. Сваки ред садржи број M и бројеве $x_1, \dots, x_M$.

Излазни подаци

У сваком од Q редова стандардног излаза исписати по један број који представља одговор на одговарајући упит.

Додатна ограничења

- Сви упити су независни.
- $N \leq 200\,000$.
- $Q \leq 30\,000$.
- Нека $\sum M$ означава збир свих вредности бројева M у датом примеру.
- $\sum M \leq 300\,000$.

#	Поени	Додатна ограничења
1	2	$M = 1$
2	12	$N \leq 1000, Q \leq 1000$
3	6	Радији i и $i + 1$ су повезани каналом за свако $1 \leq i < N$.
4	11	Највише један радио је повезан са три или више радија.
5	19	$M \leq 2$
6	13	$M \leq 3$
7	18	$M \leq 10$
8	19	Без додатних ограничења.

Примери

Улаз	Излаз
6 1 2 1 3 1 4 2 5 1 6 5 3 5 4 6 5 6 3 4 1 5 3 3 5 1 4 3 6 2 1 2 4 6	4 5 5 6 2
7 1 2 2 3 3 4 4 5 5 6 6 7 3 3 1 2 5 4 7 6 1 5 5 1 2 3 5 6	7 6 6

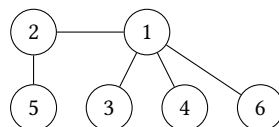
Улаз	Израз
15	15
1 2	10
2 3	13
2 4	12
2 5	2
1 6	14
3 7	12
2 8	15
2 9	13
3 10	11
3 11	15
11 12	15
10 13	12
13 14	10
13 15	13
15	
13 3 6 9 2 11 8 5 14 13 15 10 4 1	
10 8 9 5 11 2 10 6 12 15 4	
11 6 13 4 12 9 11 7 2 3 15 5	
11 14 2 15 6 7 12 9 8 3 10 4	
2 4 5	
10 2 5 7 10 8 15 9 1 14 3	
6 14 1 12 15 10 2	
3 3 11 15	
10 15 3 8 4 6 5 2 11 14 9	
4 5 9 4 7	
11 12 5 2 3 14 8 7 4 6 1 11	
12 8 13 11 3 4 9 7 1 10 2 12 14	
4 10 9 6 5	
8 7 1 8 5 12 10 15 4	
5 11 3 14 10 6	

Објашњење примера

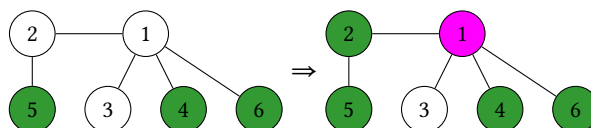
На наредним сликама су необојени чворови у стању NS, чворови обојени **зеленом бојом** су у стању S, а чворови обојени **розе бојом** су у стању D.

Први пример

Стабло у овом примеру је

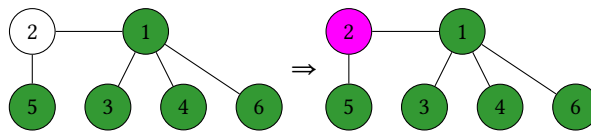


Први упит. У овом примеру, стабло пролази кроз следећи низ стања:



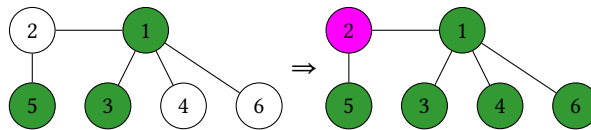
Одговор је 4, збор чворова 2, 4, 5, 6.

Други упит. У овом примеру, стабло пролази кроз следећи низ стања:



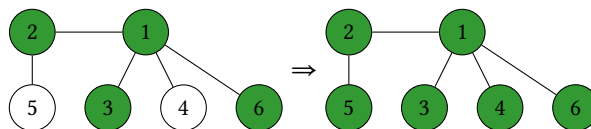
Одговор је 5, због чворова 1, 3, 4, 5, 6.

Трећи упит. У овом примеру, стабло пролази кроз следећи низ стања:



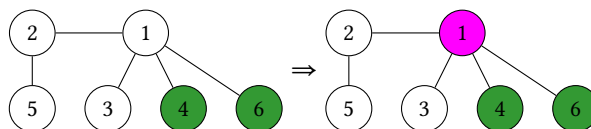
Одговор је 5, због чворова 1, 3, 4, 5, 6.

Четврти упит. У овом примеру, стабло пролази кроз следећи низ стања:



Одговор је 6, због чворова 1, 2, 3, 4, 5, 6.

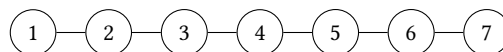
Fifth упит. У овом примеру, стабло пролази кроз следећи низ стања:



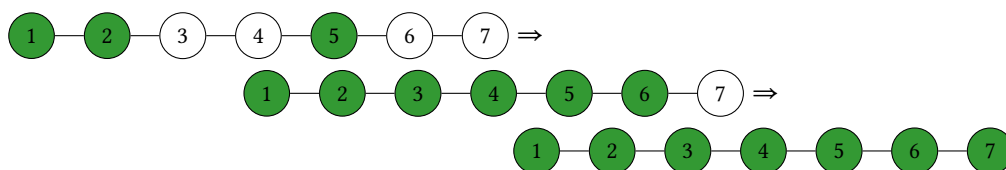
Одговор је 2, због чворова 4, 6.

Други пример

Стабло у овом примеру је

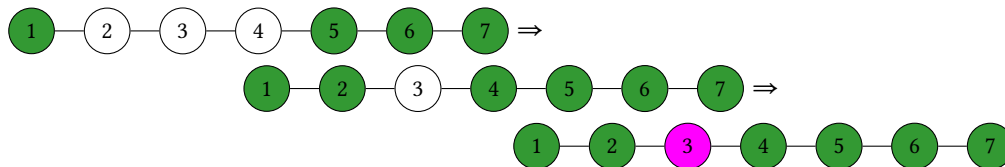


Први упит. У овом примеру, стабло пролази кроз следећи низ стања:



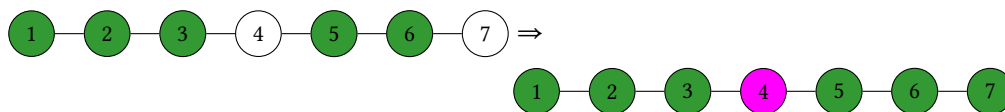
Одговор је 7, због чворова 1, 2, 3, 4, 5, 6, 7.

Други упит. У овом примеру, стабло пролази кроз следећи низ стања:



Одговор је 6, због чворова 1, 2, 4, 5, 6, 7.

Трећи упит. У овом примеру, стабло пролази кроз следећи низ стања:



Одговор је 6, због чворова 1, 2, 3, 5, 6, 7.