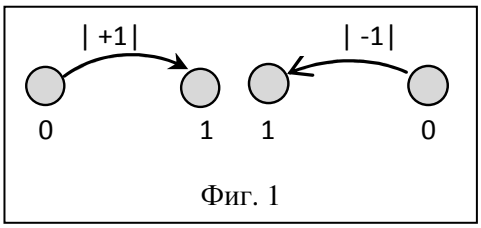
АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ТЕЛЕПОРТ

Първо ще докажем, че за всяко n и всяко k има решение с максимална възможна дължина 2^n-1. После ще посочим алгоритъм, който при зададени n и k генерира бързо желаната последователност от числа.

Нека мислим за планетите, като за 2^n върха на граф, разположени в линейна наредба и номерирани с 0, 1, …, 2^n-1. Иска се да построим ориентиран граф без цикли (даг) върху тези върхове, който представлява един маршрут, минаващ точно по веднъж през всеки връх, започвайки от връх k, за произволно дадено k, където $0 \le k \le 2^n-1$. Нещо повече, за всяко ребро (i,j) дефинираме неговата дължина $|i-j|$. Иска се всички дължини 1, 2, …, 2^n-1 да се срещат точно по един път. От общи съображения не е очевидно, че такъв граф има.



Фиг. 1

Ако върховете са само два, тоест $n = 1$, двете възможни решения са показани на Фиг. 1.Това е базата на индуктивната ни конструкция. Сега ще опишем индуктивната спъпка.

Да допуснем, че за произволно n разполагаме с всички 2^n решения, всяко от които е представено, като на Фиг. 1, чрез граф с 2^n върха.

За $n+1$ решенията са 2^{n+1} на брой, всяко с 2^{n+1} върха, които конструираме със следната процедура. Вземаме последователно по едно решение за n (с 2^n върха), правим две негови копия, да ги наречем A и B, и „вмъкваме“ A насред B. Тъй като броят върхове на B е четно число, средата е добре дефинирана. Това вмъкване води до увеличаване на дължината на всяко ребро на B с 2^n. Получава се даг с два минимални и два максимални елемента, съставен от два независими маршрута. Има точно два начина, по които можем да свържем края на единия маршрут с началото на другия. По който и да е начин да сторим това ние добавяме още едно ребро с дължина точно 2^n (Май ще трябва да се докаже – не е така очевидно както удължането на всяко ребро на външния даг!) И така, вмъквайки B в A, правим две копия от резултата и в едното копие свързваме края на A с началото на B с ребро сдължина 2^n, а в другото – края на B с началото на A ребро сдължина 2^n. Така, от всяко решение с 2^n върха, получаваме две решения, всяко с по 2^{n+1} върха. С индукция по n се доказва, че:

- в двете решения с по 2^{n+1} върха, всяка от дължините на ребра 1, 2, …, $2^{n+1}-1$ се среща точно по един път;
- получихме $2.2^n = 2^{n+1}$ графа с по 2^{n+1} върха, като всеки връх е начален връх на маршрута в някой от графите.

$n=1$		$n=2$				$n=3$							
0	1	00	01	10	11	000	001	010	011	100	101	110	111
1	0	11	10	01	00	111	110	101	100	011	010	001	000
		01	00	11	10	001	000	011	010	101	100	111	110
		10	11	00	01	110	111	100	101	010	011	000	001
						010	011	000	001	110	111	100	101

						101	100	111	110	001	000	011	010
						011	010	001	000	111	110	101	100
						110	101	110	111	000	001	010	011

Таблица 1. Всички възможни решения за малко n

Сега ще обясним как да се генерират номерата на върховете по маршрут, ако знаем n и k. Първо да разгледаме Таблица 1 с всички маршрути за n =1, 2 и 3. Всички възможни начални върхове са най-горните стойности в колоните, а всяка колона описва отгоре надолу върховете на маршрута. Таблицата е построена като се прилага горе посочения алторитъм. Такова построение може да се използва за решаване на задачата за неголеми n. Да означим с $N = 2^n$ размера на изхода на задачата. Тази задача е от особените, при които **сложността трябва да се измерва като функция не на размера на входа, а на размера на изхода!!!** Тъй като тривиалното решение изисква построяването на всички 2^n редици с 2^n елемента, сложността на тривиалния алгоритъм ще бъде $O(N^2)$.

Имаме идея за нещо като $O(N.\log N)$ но още не е написана – затова и не знаем каква ще бъде точно сложността в този случай!!!

Ще опишем едно възможно решение на задачата със сложност $O(N^2)$. Веднага виждаме, че таблиците са дуално-огледални (??? Не е е общоприет терминът!!!) спрямо вертикала по средата. Нека колоните са номерирани с числата, записани в двоична бройна система в най-горния ред. Всяка колона от жълтата част е равна на точно една колона от бялата част след битова инверсия и обратното.

Освен това, половината от редовете са излишни в смисъл, че могат да се получат от другата половина. Ако дефинираме, че *нечетните редове* са тези, в които върховете започват с нули, а *четните редове* са тези, в които започват с единици, виждаме, че всеки четен ред е битовата инверсия на нечетния ред отгоре. Наистина, не е трудно да се докаже по индукция, че маршрутите, които строим, имат свойството щото всеки втори връх има номер-битова инверсия на върха преди него.

Нека си представим всяка от таблиците с четните редове премахнати:

n = 1		n = 2				n = 3							
0	1	00	01	10	11	000	001	010	011	100	101	110	111
		01	00	11	10	001	000	011	010	101	100	111	110
						010	011	000	001	110	111	100	101
						011	010	001	000	111	110	101	100

В тези кондезирани таблици гледаме как във всяка колона се получава следващият вектор отгоре надолу. По-точно, гледаме в кои позиции има битова инверсия. Да кажем, че най-дясната позиция е номер 1 и т.н. При n = 1 имаме тривиалните случаи 0 и 1, тоест нямаме такива битови инверсии, защото кондензираната таблица има само един ред . При $n=2$ кондензираната таблица е с два реда и четири колони. По колони имаме следните битови инверсии 00 → 01, 01 → 00, 10 → 11, 11 → 10. За всяка от четирите колони битовите инверсии (става дума само за една инверсия, защото n = 2 е твърде малко) може да се опишат с един вектор на битовите инверсии, който се състои от само едно число: **1**. Това е така, защото имаме промяна само в позиция номер едно, младшият бит.

Сега да видим *n=3*. Осемте колони на кондензираната таблица са (пак написани хоризонтално):

000→001→010→011 001→000→011→010 010→011→000→001 011→010→001→000

100→101→110→111 101→100→111→110 110→111→100→101 111→110→101→100

За всяка колона, преминаването от един вектор в следващия може да се опише чрез вектора на битовите инверсии **1 2 1** . Имаме да направим три прехода, за всеки преход е достатъчно да кажем колко позиции се инвертират, защото инвертираните позиции не са пръснати произволно, а са един подвектор, „опиращ се" в дясната част.

Не би трябвало да е трудно да се покаже, че за n=4 векторът на промените е **1 2 1 3 1 2 1** и че изобщо векторът на промените при *n* бита дума на Зимин Z_{n-1} (Не знам кой е Зимин – аз бих казал че това е векторът от позициите, в които трябва да променим всеки получен вестор за да подредим двоичните вектори в код на Gray). Думите на Зимин (Zimin words) над азбука {1,2,3, …} се дефинират рекурсивно: $Z_1 = 1$, $Z_n = Z_{n-1}\ n\ Z_{n-1}$.

Примерно, ако *n =4* и искаме да изчислим маршрута, стартиращ от връх номер 5, в двоична бройна система 0101, кондезираната колона (тоест, нечетните върхове от маршрута) е: 0101→0100→0111→0110→0001→0000→0011→0010

откъдето виждаме, че маршрутът е, отляво надясно,

0101 1010 0100 1011 0111 1000 0110 1001 0001 1110 0000 1111 0011 1100 0010 1101

Естествено, примерът не е доказателство, че думите на Зимин описват местата за инверсиите…Това е програмата, която имплементира линейния алгоритъм. Пресмятането на числата от думите на Зимин (или променящите се позиции при кода на Грей, което е едно и също) може да стане с произволна друга таблица с размер степен на двойката без скоростта да пострада съществено.

Програма:

```
#include <stdio.h>
unsigned grayc(int i)
{  unsigned gray[256] =
   { 1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,5,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,6,
     1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,5,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,7,
     1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,5,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,6,
     1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,5,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,8,
     1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,5,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,6,
     1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,5,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,7,
     1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,5,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,6,
     1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,5,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,9};
   unsigned x,y=i%256;
   if(y!=255) return gray[y];
   x=0; while(i>0&&i%256==255) {i/=256;x++;}
   return x*8+gray[i%256];
}
int main()
{  int N,K;
   // could be extended to N
   int invert[21]={0,1,3,7,15,31,63,127,255,511,1023,2047,4095,8191,
```

```
                16383,32767,65535,131071,262143,524287,1048575};
   unsigned prev,curr,next;  //three consecutive
   scanf("%d %d",&N,&K);
   printf("%d\n",invert[N]); //coincidence
   prev=K;                   // start planet
   for(int i=0;i<invert[N]-2;i=i+2)
   {
      curr=prev^invert[N];          // invert all N bits
      next=prev^invert[grayc(i/2)]; // invert bits defined from Gray
      if(curr>prev) printf("%d ",curr-prev);
      else printf("-%d ",prev-curr);
      if(next>curr) printf("%d ",next-curr);
      else printf("-%d ",curr-next);
      prev=next;
   }
   //last step - only first inversion
   curr=prev^invert[N];
   if(curr>prev) printf("%d\n",curr-prev);
   else printf("-%d\n",prev-curr);
   return 0;
}
```

Автори: Красимир Манев и Минко Марков