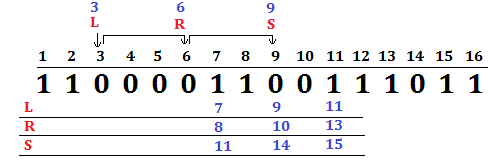
**ДЕТСКА ГРАДИНА 2**

Явно 2≤M≤N и 1≤K≤N/2.

I начин: С два вложени цикъла се намират M и K. В зависимост от реализацията на обхождане на редицата А, решението ще работи за ХХХ% от тестовете.

II начин: Разделяме редицата по групи от единици и нули. Използваме структура, подобна на списък, спрямо първите елементи на всяка група:



Например за групата от нули в интервала [3;6] въвеждаме L и R – начало и край на интервала, и S – номера на следващата група от нули. По този начин може да прескачаме елементи от редицата А и да сменяме единици с нули. Използваме масив B[] от структура, в която има тези 3 компонента R, L и S.

Нека сме стигнали до I=3, A[I]=A[3]=0 и пробваме за K=3. Алгоритъмът е следния:

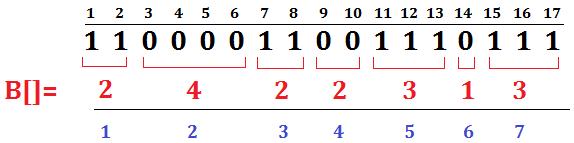
Нека Sum е броят на еднаквите цифри. Когато започваме от някой интервал, нулираме Sum. Дължината на интервала е Sum=Sum+B[3].R-B[3].L+1=0+6-3+1=4 > K. Тогава сменяме нулите с единици, като знаем, че следващата поредица започва от B[3].R+1, т.е. I=B[3].R+1=6+1=7.

Sum=0. Дължината на поредицата от 1, започваща в I=7 е Sum=Sum+B[7].R-B[7].L+1=0+8-7+1=2 < K. Тогава отиваме в следващия интервал от същия тип, т.е. от единици. Той започва в B[7].S=11, правим I=B[7].S=11.

Броят на единиците с начало 11 е Sum=Sum+B[11].R-B[11].L+1=2+3=5, което е по-голямо от K. Тогава сменяме единиците с нули, като правим I=B[11].R+1=14.

Всеки път към някакъв брояч Br, когато S≥K, прибавяме K. Накрая Br ни дава максималния брой деца при текущото K.

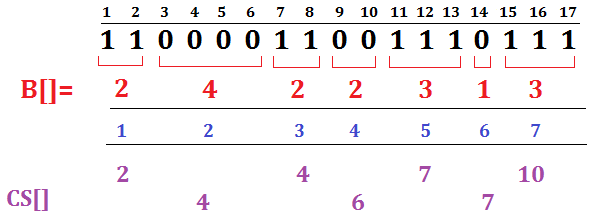
III начин: Той използва идеята на II начин, но сега масивът B[] няма да е от тип структура. Първо да отбележим, че нищо не се променя, ако всяка 1 стане 0 и всяка 0 стане 1. Отговорът за редиците 0110101 и 1001010 е един и същ. Тогава ще считаме, че първият елемент е 1.



Групираме единици и нули и масивът В е броят им в групата: B[1]=1, защото първо има две 1, след това В[2]=4, защото следват четири н и т.н.

Аналогично на II начин изчисляваме отговорите за всяко K, като за K=1 отговорът е броя на елементите на B[]. Единственото подобрение е, че с двоично търсене намираме веднага за текущия номер кой е следващия номер, при който има K поредни символа.

За целта правим масив CS[] от частични суми на A[i], обаче по четни и нечетни индекси на масива B[]:



В лилаво е целия масив CS[], като на горния ред са частичните суми за единиците, т.е. това са нечетните индекси на CS[], а на долния ред са частичните суми за нулите - четните индекси на CS[], като /CS[0]=0.

Тогава в двоичното търсене трябва да се редува търсене по четни и нечетни индекси и то през един, т.е. със стъпка 2. Намираме ТС – най-големия четен индекс на B[] и най-големия му нечетен индекс TN.

Нека К=3 и започваме от номер 1 в масива B[], т.е. I=1. Понеже I е нечетно число, ни интересуват нечетните позиции на CS[], които пазят частичните суми на единиците. Търсим в тези нечетни позиции в интервала [1;TN] най-малкото число CS[P], за което CS[P] ≥ CS[I-2]+K. Понеже I – 2 = 1 – 2 = -1<0, вземаме индекс 0 и се получава CS[0]+K=0+3=3. Отговорът от двоичното търсене нека е позиция Р, в случая P=3, т.е. CS[P]=CS[3]=4. И наистина, ако започнем от I=1, то B[1]=2 < 3, добавяме следващото B[3]=2 и се получава 4>К=3.

Преместваме се на позиция I=P+1=4. Това е четна позиция, значи започват нулите. Търсим в интервала [4,TC] числото CS[I-2]+K=C[2]+3=7. Отговорът е в позиция P=6, и в случая е точно: CS[6]=7.

Правим I=P+1=7 и сега ще търсим в [7;TN] числото CS[I-2]+K=7+3=10. В случая CS[7]=10. Правим I=P+1=8 и търсим в [8;TС] числото CS[I-2]+K=7+3=10 и т.н.