Problem

An alien robot is threatening the universe, using a beam that will destroy all algorithms knowledge. We have to stop it!

Fortunately, we understand how the robot works. It starts off with a beam with a strength of 1, and it will run a program that is a series of instructions, which will be executed one at a time, in left to right order. Each instruction is of one of the following two types:

- C (for "charge"): Double the beam's strength.
- S (for "shoot"): Shoot the beam, doing damage equal to the beam's current strength.

For example, if the robot's program is SCCSSC, the robot will do the following when the program runs:

- Shoot the beam, doing 1 damage.
- Charge the beam, doubling the beam's strength to 2.
- Charge the beam, doubling the beam's strength to 4.
- Shoot the beam, doing 4 damage.
- Shoot the beam, doing 4 damage.
- Charge the beam, increasing the beam's strength to 8.

In that case, the program would do a total of 9 damage.

The universe's top algorithmists have developed a shield that can withstand a maximum total of **D** damage. But the robot's current program might do more damage than that when it runs.

The President of the Universe has volunteered to fly into space to hack the robot's program before the robot runs it. The only way the President can hack (without the robot noticing) is by swapping two adjacent instructions. For example, the President could hack the above program once by swapping the third and fourth instructions to make it SCSCSC. This would reduce the total damage to 7. Then, for example, the president could hack the program again to make it SCSSCC, reducing the damage to 5, and so on.

To prevent the robot from getting too suspicious, the President does not want to hack too many times. What is this smallest possible number of hacks which will ensure that the program does no more than **D** total damage, if it is possible to do so?

Consider Example Programm P: SCSCCSC SS

Cmd	S	C	S	C	C	S	C	S	S
Damage	2^0	-	2^1	-	-	2^3	-	2^4	2^4

$\Rightarrow$ Damage $D = \sum_{k=0}^{n} a_k 2^k$

where n is the number of C's in P and a_k the number of S commands with k C's in front.

$\Rightarrow$ P can be written as

k	0	1	2	3	4
a_k	1	1	0	1	2

and can be backtransformed as well

$\Rightarrow$

k	0	1	2	3	4
a_k	1	1	0	1	2

$\Rightarrow$ "S" + "C" + "S" + "C" + " " + "C" + "S" + "C" + "SS"

$\Rightarrow$ SCSCCSCSS

Looking at Neighbour swapping:

Consider $S \leftrightarrow S$ bzw $C \leftrightarrow C \Rightarrow$ Sequence doesn't change $\Rightarrow$ Damage doesn't change

consider $C \leftrightarrow S \Rightarrow$ e.g. SC $\Rightarrow D = 1 \cdot 2^0 + 0 \cdot 2^1 = 1$

$\Rightarrow$ CS $\Rightarrow D = 0 \cdot 2^0 + 1 \cdot 2^1 = 2$

$\Rightarrow$ swapping an S with k C's in front

A: to the Left $\Rightarrow$ new S has $k-1$ C's in front

B to the Right $\Rightarrow$ $k+1$ C's in front

$\Rightarrow$ $D_{left} = a_0 2^0 + \dots + (a_{k-1}+1) 2^{k-1} + (a_k - 1) 2^k + \dots + a_n 2^n$
$D_{right} = a_0 2^0 + \dots + a_k 2^k + (a_{k+1}+1) 2^{k+1} + \dots + a_n 2^n$

$\Rightarrow$ because $(a_{k-1}+1) 2^{k-1} + (a_k - 1) 2^k = a_{k-1} 2^{k-1} + 2^{k-1} + 2^k a_k - 2^k$

$= a_{k-1}2^{k-1} + a_k 2^k + (1-2)2^{k-1} < a_{k-1}2^{k-1} + a_k 2^k$

$\Rightarrow D_{left} < D_{before}$ $\quad$ $D_{before} < D_{right}$ follows similarly

Goal: Decreasing D $\quad$ Approach: Greedy

as $D_{left} < D_{before}$ or more precise $D_{left} - 2^{k-1} = D_{right}$

a maximum decrease happens for a maximum value of k

Example $P = SCSCCSCSS$

k	0	1	2	3	4
a_k	1	1	0	1	2

$D = 1+2+0+8+32 = 43$

$\Rightarrow$ $P = SCSCCSSCS$

k	0	1	2	3	4
a_k	1	1	0	2	1

$D = 1+2+0+16+16 = \underline{\underline{35}}$

$k=4$

$\Rightarrow D_{new} = 43 - 2^{4-1} = \underline{\underline{35}}$

SCSCCSSSC

$\Rightarrow$

k	0	1	2	3	4
a_k	1	1	0	3	0

$D = 35 - 8 = 27$

$\Rightarrow$ ... as $\sum a_k$ has to stay constant D can be decreased up to an maximum of

k	0	1	2	3	4
a_k	5	0	0	0	0

$\Rightarrow D_{min} = \sum a_k$