

~~$\phi_{21}$~~

$n C_3$

[You have some sticks]

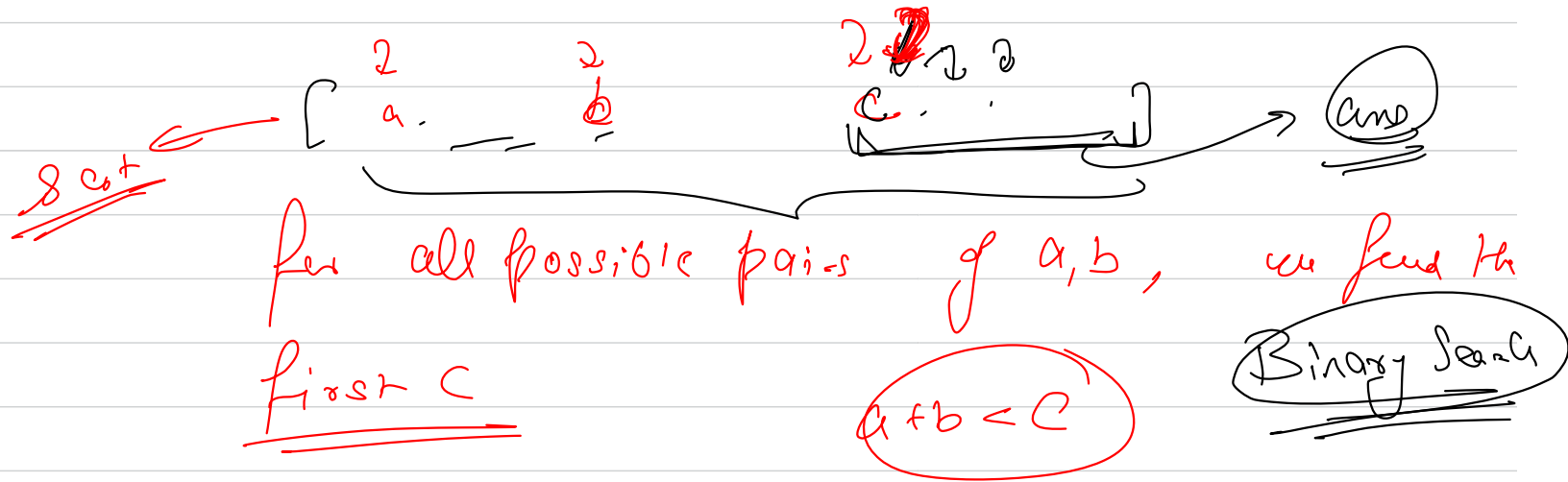
You can pick any 3

order  
doesn't  
matter

We know for any 2 sides  $a, b$  if we  
have a 3<sup>rd</sup> side  $c$  such that

$a + b < c \longrightarrow$  1 possible ans

That for any 2 sides  $a, b$ , if we find all the possible  $c$ 's such that  $a + b < c$ , then we will have all those as our ans.



Petya has the word  $t$ , he wants to make the word  $p$  from it. Petya begins to delete the letters in a certain order, which is described as a permutation of indices of the letters of the word  $t$ :  $a_1 \dots a_{|t|}$ . Note that after deleting a letter, the numbering does not change.

His brother Vasya is afraid that Petya may delete too many letters, so he will not get the word  $p$  in the end. Vasya's task is to stop his brother at some point and finish deleting himself in such a way, that the resulting word will be  $p$ . Since Petya likes this activity, Vasya wants to stop him as late as possible. Your task is to tell how many letters Petya can delete out before Vasya stops him.

It is guaranteed that the word  $p$  can be obtained by deleting letters from  $t$ .

### Input

The first and second lines of the input file contain the words  $t$  and  $p$ , respectively. Words consist of lowercase letters of the Latin alphabet ( $1 \leq |p| < |t| \leq 200\,000$ ).

The next line contains the permutation  $a_1 \dots a_{|t|}$  of letter indices, which specifies the order in which Petya deletes the letters of the word  $t$  ( $1 \leq a_i \leq |t|$ , all  $a_i$  are different).

### Output

Print one number, the maximum number of letters that Petya can delete.

1 1 2 3 4 5 6 7  
 $t = ababcbab$

2  
 $p = abb$

$[5, 3, 4, 1, 7, 6, 2]$   
 ↑

ans → 3

freq

a →  
 b →  
 c →

no of operation

0

Search space  
 $de(t) - de(p) + 1$   
delete  
mid

1 2 3 4 5 6 7  
a b a b c b a → t

abb → f

[5, 3, 4, 1, 7, 6, 2]

(5, 3, 4, 1)

mid = 4

mid = 3

new string

new

bbq

O(n)

[5, 3, 4, 1]

O(1)

new → b b q

f → a b b

new → a b b q

f → a b b

→ of let's say we have mid no. of operation  
// available, so we prepare a Set of  
// size mid, and add the first mid operations

→ then iterate over the string 's' & create a  
new string which will only include character  
present on the indices not in the Set.



$[1, 2, 3]$

$[2, 3]$

$\{3, 2\}$  subset