Problem B. Brackets

Time limit: 3 seconds
Memory limit: 1024 megabytes

Originally, there were only three kinds of brackets. But due to the harsh realities of life, the greater-than and less-than signs started moonlighting as angle brackets.

—Stir-fried-chicken

Betty has a bracket sequence s of length n, which consists of eight types of brackets: "`()[]{}<>`". Additionally, she has m sub-bracket sequences, and the i-th sub-bracket sequence t_i is the sequence formed by concatenating the characters from the l_i-th to the r_i-th position of s, i.e., $t_i = s_{l_i} s_{l_i+1} \cdots s_{r_i}$.

Betty wants to know, by taking out several pairs from these m sub-bracket sequences and concatenating sequences in the same pair, how many valid bracket sequences† can be formed at most? Formally, you need to find as many pairs (a_i, b_i) as possible such that:

1. For any integer x from 1 to m, the sum of its occurrences in a and its occurrences in b should be no larger than 1.
2. For any pair (a_i, b_i), $t_{a_i} t_{b_i}$ is a valid bracket sequence†.

† A bracket sequence x is valid if and only if it satisfies one of the following conditions:

- $|x| = 0$, i.e., x is an empty string.
- $x = LyR$, where y is a valid bracket sequence, and L, R are left and right brackets of the same type (i.e., $L =$ "`(`" and $R =$ "`)`", or $L =$ "`[`" and $R =$ "`]`", or $L =$ "`{`" and $R =$ "`}`", or $L =$ "`<`" and $R =$ "`>`").
- $x = y_1 y_2$, where both y_1 and y_2 are valid bracket sequences.

Input

Each test file contains multiple test cases. The first line contains the number of test cases T ($1 \le T \le 10^4$). The description of the test cases follows.

The first line contains two integers n and m ($1 \le n, m \le 5 \times 10^5$), which represent the length of s and the number of sub-bracket sequences, respectively.

The second line contains a string s ($|s| = n$), which consists only of the eight brackets "`()[]{}<>`".

The next m lines each contain two integers l_i and r_i ($1 \le l_i \le r_i \le n$).

For each test file, it is guaranteed that the sum of n over all test cases does not exceed 5×10^5, and the sum of m over all test cases does not exceed 5×10^5.

Output

For each test case, output a single integer on a new line, representing the maximum number of valid bracket sequences that can be obtained by concatenating pairs of sub-bracket sequences.

Example

standard input	standard output
4 8 1 ()[]{}<> 3 6 2 6)(1 1 1 1 1 1 2 2 2 2 2 2 6 2 ([)(]) 1 3 4 6 22 8 ([{}<<<<])>>>>([]){()} 3 8 11 14 1 10 3 4 19 22 20 21 17 20 21 22	0 3 0 2

Note

In the first test case, t_1 = "[]{}". Although t_1 is already a valid bracket sequence, there are no other sub-bracket sequences that can be paired with it, so the output is 0.

In the second test case, there are six sub-bracket sequences containing three "(" and three ")", which can be used to form three valid bracket sequences "()". Therefore, the output should be 3.

In the third test case, no valid bracket sequences can be formed.

In the fourth test case, one possible matching is t_1t_2: "{}««" + "»»", and t_4t_5: "{}" + "{()}".