Pair and unpair weightest string

Once I went market to buy a teddy. The shopkepper offered a puzzle and said he'll give me whatever i wish at a huge discount if i give him the working piece of code for following problem.

   You will be given a string S of length N, containing either 'A' or 'B'. Make all the possible lists from the given list, with the given operation that you are allowed to change one symbol into another. Then assign maximum weight to each of the string and get the string with maximum weight && return the weight.

You have only following two ways to assign weight to symbol :
1. Pair a S[i] symbol with adjacent one (S[i] or S[i-1]). Each symbol can only be paired with exactly one symbol. Weight of a pair is 4. Pair will be either 'AB' or 'BA'.
2. Keep a S[i] Symbol indepedent. Weight of unpaired symbol is 1.
3. Each symbol is either involved in exactly one either pair or unpaired status.
4. For each string, Keep in mind the number of symbols(K) with index i, 1<=i<=N, which are changed from original symbol S[i].
5. For each of the possible lists, Add all the weight assigned as a pair and unpair, then subtract K from it. assign this final value as string weight. Add each paired weight once exactly.

The Task is to maximize the weight of the string. Suppose S="ABB", possible transforms can be {"ABB", "ABA", "AAB", "AAA", "BBB", "BBA", "BAB", "BAA"} with respective weight {5,4,4,1,2,3,3,2}. Maximum weight is found 5 in string 1.

Constraints: 1<=N<=10^5, 1<=T<=500.

Input :
First line contains t, number of testcases. For each testcase, first line contains N, length of the string. Second line contains string itself.
Output:
For Each testcase, Output maximum weight of a string that can be obtained.

Example
Input:
5
3
ABB
7
ABBBAAA
8
BAAAAAAA
12
AAAAAAAAAAAA
11
ABABBBABABA

Output:

5
12
13
18
21