Jacek Tomasiewicz **Marek Sommer, Paweł Wegner**
Treść zadania, Opracowanie Program

XXII OI, etap II, dzień drugi, 12.02.2015

Trzy wieże

Bitoni uwielbia się bawić. W swoim pokoju ułożył w jednym rzędzie n klocków. Każdy z klocków ma jeden z trzech kolorów: biały, szary lub czarny. Bitoni chciałby wybrać pewien spójny fragment rzędu klocków, a następnie z klocków z tego fragmentu zbudować wieże.

Każda wieża może się składać z klocków tylko jednego koloru i nie może być dwóch wież o tym samym kolorze (zatem Bitoni zbuduje co najwyżej trzy wieże). Ponadto nie może być dwóch wież o tej samej wysokości (tzn. każda wieża musi być zbudowana z innej liczby klocków niż pozostałe). Zakładamy, że Bitoni musi wykorzystać wszystkie wybrane przez siebie klocki. Pomóż Bitoniemu i napisz program, który znajdzie najdłuższy fragment rzędu klocków spełniający jego wymagania.

Wejście

Pierwszy wiersz standardowego wejścia zawiera jedną liczbę całkowitą n ($1 \leqslant n \leqslant 1\,000\,000$), oznaczającą liczbę klocków. Kolejny wiersz zawiera napis złożony z n liter $a_1 a_2 \ldots a_n$, w którym a_i jest jedną z liter `B`, `S` *lub* `C` *i oznacza kolor i-tego klocka w rzędzie (litera* `B` *oznacza klocek koloru białego, litera* `S` *klocek szary, a litera* `C` *klocek czarny).*

W testach wartych 30% punktów zachodzi dodatkowy warunek $n \leqslant 2500$.

Wyjście

Pierwszy i jedyny wiersz standardowego wyjścia powinien zawierać jedną liczbę całkowitą, równą liczbie klocków w najdłuższym spójnym fragmencie rzędu, który spełni wymagania Bitoniego.

Przykład

Dla danych wejściowych:
```
9
CBBSSBCSC
```

poprawnym wynikiem jest:
```
6
```

Wyjaśnienie do przykładu: *Bitoni może wybrać fragment złożony z 6 klocków:* `BSSBCS`, *z których zbuduje szarą wieżę złożoną z trzech klocków, białą z dwóch klocków oraz czarną z jednego klocka.*

Testy „ocen”:

- `1ocen`*: $n = 2500$, rząd klocków jest następujący:* $\mathtt{B}^{1248}\mathtt{C}\underline{\mathtt{SB}^{1250}}$ *(napis $\mathtt{B}^k$ oznacza k-krotne powtórzenie litery* `B`*); najdłuższy fragment rzędu klocków, który Bitoni może wybrać, został podkreślony;*
- `2ocen`*: $n = 1\,000\,000$, rząd klocków jest okresowy:* `BSCBSCBSC...BSCBSCB`*; Bitoni może zbudować tylko jedną wieżę z jednego klocka.*

Rozwiązanie

W zadaniu mamy dane słowo Z o długości n składające się z literek `A`, `B`, `C` (dla czytelności opisu zamiast literki `S` używamy literki `A`) reprezentujących kolory kolejnych klocków ułożonych w rzędzie. Musimy znaleźć najdłuższy fragment ciągu klocków, taki że liczba wystąpień klocków każdego koloru w tym fragmencie będzie inna (tak aby wysokości wież zbudowanych z klocków tego samego koloru były różne). Fragment (niekoniecznie najdłuższy) spełniający ten warunek nazwiemy *poprawnym*.

Rozwiązanie siłowe $O(n^3)$

Dla każdego spójnego fragmentu ciągu klocków możemy sprawdzić, czy jest on poprawny. Łatwo to wykonać w czasie liniowym: wystarczy policzyć i porównać liczbę wystąpień klocków każdego koloru. Jako że wszystkich spójnych fragmentów jest $O(n^2)$, a pojedyncze sprawdzenie zajmuje czas liniowy względem długości fragmentu, rozwiązanie to ma złożoność czasową $O(n^3)$.

Rozwiązanie to zaimplementowane jest w pliku `trzs2.cpp`. Za poprawne zaprogramowanie takiego rozwiązania na zawodach można było uzyskać około 15% punktów.

Rozwiązanie wolne $O(n^2)$

Sprawdzenie poprawności fragmentu możemy wykonać szybciej, w czasie $O(1)$, po wcześniejszym przetworzeniu ciągu klocków w czasie $O(n)$. Aby móc efektywnie obliczać liczbę wystąpień klocków każdego koloru w dowolnym fragmencie, w pierwszym kroku wyznaczymy ciąg sum częściowych słowa Z dla każdego koloru oddzielnie. Załóżmy, że rozpatrujemy kolor `A`.

Niech $w_i = 1$, jeśli $Z_i =$ `A`, zaś w przeciwnym przypadku $w_i = 0$. i-tą sumę częściową (dla $1 \leqslant i \leqslant n$) definiujemy jako $b_i = w_1 + w_2 + \ldots + w_i$, jednocześnie przyjmując $b_0 = 0$. Zauważmy, że $b_i = b_{i-1} + w_i$, więc ciąg $b_1, \ldots, b_n$ można obliczyć w czasie $O(n)$. Wartości b_i pozwalają obliczyć liczbę wystąpień klocków koloru `A` w dowolnym fragmencie klocków $Z_i, \ldots, Z_j$ w czasie stałym ze wzoru $b_j - b_{i-1}$. W analogiczny sposób możemy wyznaczyć ciąg sum częściowych dla koloru `B` i `C`.

Implementacja takiego rozwiązania znajduje się w pliku `trzs3.cpp`. Rozwiązanie tego typu otrzymywało na zawodach około 30% punktów.

Rozwiązanie wzorcowe $O(n)$

Udowodnimy, że optymalne wyniki znajdują się blisko jednego z końców ciągu klocków. Dokładniej, jeśli $[i, j]$ jest przedziałem reprezentującym optymalny fragment ciągu klocków (spośród wszystkich optymalnych wybierzmy ten o najmniejszym i), to $i \leqslant 3$ lub $j \geqslant n-2$. Jeśli tak rzeczywiście jest, to można wyznaczyć $O(n)$ kandydatów (przedziałów), wśród których znajduje się optymalny wynik. Te przedziały to $[1, j]$, $[2, j]$, $[3, j]$ oraz $[i, n]$, $[i, n-1]$, $[i, n-2]$ dla wszystkich i, j, takich że przedziały te będą poprawnie zdefiniowane – początek przedziału nie będzie większy od końca przedziału.

Po przetworzeniu ciągu w czasie $O(n)$, będziemy mogli w czasie stałym stwierdzić, czy przedział odpowiada poprawnemu fragmentowi (powiemy wtedy, że przedział jest poprawny). Ostatecznie, rozwiązanie to ma złożoność czasową $O(n)$ i zostało zaimplementowane w pliku `trz2.cpp`. Zaimplementowanie takiego rozwiązania w trakcie zawodów było nagradzane maksymalną liczbą punktów. Autorem tego rozwiązania jest Marek Sommer.

Dowód

Niech przedział $[i, j]$ będzie najdłuższym poprawnym przedziałem, a spośród wszystkich najdłuższych niech będzie tym, który ma najmniejsze i. Spróbujemy założyć, że przedział nie znajduje się blisko żadnego z końców ciągu, czyli $i > 3$ i $j < n - 2$. To oznacza, że z każdej strony przedziału znajdują się przynajmniej po trzy klocki, których nie można dołożyć do tego przedziału.

$\cdots$	?	?	?	Z_i	Z_{i+1}	$\cdots$	Z_{j-1}	Z_j	?	?	?	$\cdots$

Dowód będzie opierał się na rozważeniu wielu przypadków (ale dzięki temu nie ma ich już w algorytmie) i pokazaniu, że w każdym z nich dochodzimy do sprzeczności – pokażemy, że będzie istniał lepszy lub równoważny przedział, który znajduje się blisko jednego z końców ciągu. Najpierw jednak zachęcamy Czytelnika do próby samodzielnego przeprowadzenia dowodu, który można wykonać w łatwy sposób, rozpisując różne przypadki na kartce. Poniżej przedstawiamy sposoby rozpatrzenia poszczególnych przypadków.

W przedziale $[i, j]$ znajdują się klocki tylko jednego koloru

Mamy więc $Z_i = Z_{i+1} = \ldots = Z_{j-1} = Z_j$. Jeśli $i \neq j$, to dokładając element Z_{i-1}, dostaniemy również poprawny przedział (klocków koloru Z_i jest co najmniej 2, więc nowy klocek tego samego bądź innego koloru niczego nie popsuje). Z tego wynika, że przedział $[i, j]$ nie jest najdłuższy poprawny – dostajemy sprzeczność. Jeśli natomiast $i = j$, to przedział jest długości 1. Dowolny przedział długości 1 jest poprawny, więc poprawny jest również przedział $[1, 1]$. Otrzymujemy sprzeczność, bo przedział $[i, j]$ miał być optymalnym rozwiązaniem o najmniejszym i.

W przedziale $[i, j]$ znajdują się klocki dokładnie dwóch kolorów

W tym przypadku możemy przyjąć, że w przedziale $[i, j]$ znajdują się klocki trzech kolorów, przy czym jedna z utworzonych wież ma wysokość 0, co sprowadza się do następnego przypadku, opisanego poniżej.

W przedziale $[i, j]$ znajdują się klocki dokładnie trzech kolorów

Niech $|X|$ oznacza liczbę wystąpień klocków koloru X w przedziale $[i, j]$. Bez straty ogólności możemy założyć, że $|\mathtt{A}| < |\mathtt{B}| < |\mathtt{C}|$. Co teraz może się kryć pod znakami zapytania?

$\cdots$	$?_1$	$?_2$	$?_3$	Z_i	Z_{i+1}	$\cdots$	Z_{j-1}	Z_j	$?_4$	$?_5$	$?_6$	$\cdots$

Znaki $?_3$ i $?_4$ nie mogą być równe C (w przeciwnym przypadku można by było powiększyć nimi rozwiązanie).

Przypadek, w którym $?_3 = \texttt{B}$

$\cdots$	$?_1$	$?_2$	B	Z_i	Z_{i+1}	$\cdots$	Z_{j-1}	Z_j	$?_4$	$?_5$	$?_6$	$\cdots$

Czy $?_4$ może być równe B? Nie, ponieważ wtedy przedział $[i, j]$ sąsiadowałby z dwiema literkami B i w zależności od tego, czy $|\texttt{B}| + 1 = |\texttt{C}|$, czy nie, moglibyśmy powiększyć rozwiązanie albo przy pomocy jednej, albo dwóch literek B. Znak $?_4$ nie może być też równy C (co było powiedziane wcześniej). Zatem otrzymujemy $?_4 = \texttt{A}$.

$\cdots$	$?_1$	$?_2$	B	Z_i	Z_{i+1}	$\cdots$	Z_{j-1}	Z_j	A	$?_5$	$?_6$	$\cdots$

Wiadomo, że $|\texttt{A}| + 1 = |\texttt{B}|$ i $|\texttt{B}| + 1 = |\texttt{C}|$ (gdyby tak nie było, to można by było powiększyć rozwiązanie o jedną literkę A lub jedną literkę B). Możemy więc zapisać, że $|\texttt{A}| = x$, $|\texttt{B}| = x + 1$ i $|\texttt{C}| = x + 2$ dla pewnego x.

Czy $?_5$ może być równe C? Nie, ponieważ wtedy moglibyśmy dodać znaki $?_3$, $?_4$ i $?_5$, otrzymując lepsze rozwiązanie, gdzie $|\texttt{A}| = x + 1$, $|\texttt{B}| = x + 2$ i $|\texttt{C}| = x + 3$. Czy $?_5$ może być równe B? Nie, ponieważ wtedy moglibyśmy dodać znaki $?_3$, $?_4$ i $?_5$, otrzymując lepsze rozwiązanie, gdzie $|\texttt{A}| = x + 1$, $|\texttt{C}| = x + 2$ i $|\texttt{B}| = x + 3$. Zatem otrzymujemy $?_5 = \texttt{A}$.

$\cdots$	$?_1$	$?_2$	B	Z_i	Z_{i+1}	$\cdots$	Z_{j-1}	Z_j	A	A	$?_6$	$\cdots$

$?_6$ nie może być równe A ani C, ponieważ dodając znaki $?_4$, $?_5$ i $?_6$, otrzymalibyśmy lepsze rozwiązanie. Zatem otrzymujemy $?_6 = \texttt{B}$.

$\cdots$	$?_1$	$?_2$	B	Z_i	Z_{i+1}	$\cdots$	Z_{j-1}	Z_j	A	A	B	$\cdots$

$?_2$ nie może być równe B ani C, ponieważ dodając znaki $?_2$, $?_3$ i $?_4$, otrzymalibyśmy lepsze rozwiązanie. Zatem otrzymujemy $?_2 = \texttt{A}$.

$\cdots$	$?_1$	A	B	Z_i	Z_{i+1}	$\cdots$	Z_{j-1}	Z_j	A	A	B	$\cdots$

$?_1$ nie może być równe B ani C, ponieważ dodając znaki $?_1$, $?_2$ i $?_3$, otrzymalibyśmy lepsze rozwiązanie. Zatem otrzymujemy $?_1 = \texttt{A}$.

$\cdots$	A	A	B	Z_i	Z_{i+1}	$\cdots$	Z_{j-1}	Z_j	A	A	B	$\cdots$

Okazuje się, że dodając wszystkie znaki od $?_1$ do $?_6$, otrzymalibyśmy lepsze rozwiązanie, w którym $|\texttt{C}| = x + 2$, $|\texttt{B}| = x + 3$ i $|\texttt{A}| = x + 4$, więc mamy sprzeczność.

Przypadek, w którym $?_3 = \mathtt{A}$

$\cdots$	$?_1$	$?_2$	A	Z_i	Z_{i+1}	$\cdots$	Z_{j-1}	Z_j	$?_4$	$?_5$	$?_6$	$\cdots$

Gdyby $?_4 = \mathtt{B}$, to otrzymalibyśmy przypadek symetryczny do poprzedniego przypadku, w którym $?_3 = \mathtt{B}$ (który był sprzeczny), więc $?_4$ nie może być równe B ani też C (co było powiedziane wcześniej). Zatem otrzymujemy $?_4 = \mathtt{A}$.

$\cdots$	$?_1$	$?_2$	A	Z_i	Z_{i+1}	$\cdots$	Z_{j-1}	Z_j	A	$?_5$	$?_6$	$\cdots$

Wiadomo, że $|\mathtt{A}| + 1 = |\mathtt{B}|$ i $|\mathtt{A}| + 2 = |\mathtt{C}|$ (gdyby tak nie było, to można by było dodać jedną lub dwie literki A, otrzymując lepsze rozwiązanie). Możemy więc zapisać, że $|\mathtt{A}| = x$, $|\mathtt{B}| = x + 1$ i $|\mathtt{C}| = x + 2$, dla pewnego x.

$?_5$ nie może być równe A ani C, ponieważ dodając znaki $?_3$, $?_4$ i $?_5$, otrzymalibyśmy lepsze rozwiązanie. Zatem otrzymujemy $?_5 = \mathtt{B}$.

$\cdots$	$?_1$	$?_2$	A	Z_i	Z_{i+1}	$\cdots$	Z_{j-1}	Z_j	A	B	$?_6$	$\cdots$

Znak $?_2$ jest w tym przypadku symetryczny do $?_5$, więc tak samo możemy wywnioskować, że $?_2 = \mathtt{B}$.

$\cdots$	$?_1$	B	A	Z_i	Z_{i+1}	$\cdots$	Z_{j-1}	Z_j	A	B	$?_6$	$\cdots$

$?_6$ nie może być równe B ani C, ponieważ dodając znaki $?_4$, $?_5$ i $?_6$, otrzymalibyśmy lepsze rozwiązanie. Zatem otrzymujemy $?_6 = \mathtt{A}$.

$\cdots$	$?_1$	B	A	Z_i	Z_{i+1}	$\cdots$	Z_{j-1}	Z_j	A	B	A	$\cdots$

Znak $?_1$ jest w tym przypadku symetryczny do $?_6$, więc tak samo możemy wywnioskować, że $?_1 = \mathtt{A}$.

$\cdots$	A	B	A	Z_i	Z_{i+1}	$\cdots$	Z_{j-1}	Z_j	A	B	A	$\cdots$

Okazuje się, że dodając wszystkie znaki od $?_1$ do $?_6$, otrzymalibyśmy lepsze rozwiązanie, w którym $|\mathtt{C}| = x + 2$, $|\mathtt{B}| = x + 3$ i $|\mathtt{A}| = x + 4$, więc mamy sprzeczność.

Mniejsza liczba kandydatów

Udowodniliśmy, że wystarczy sprawdzić przedziały, które są w odległości nie większej niż 2 od któregoś z końców ciągu klocków. Jest to najmniejsze możliwe ograniczenie. Gdyby algorytm sprawdzał tylko przedziały w odległości nie większej niż 1, to źle odpowiedziałby w następującym przypadku:

$\texttt{AB}\underline{\texttt{CACBAC}}\texttt{BA}$

Nie wystarczy również sprawdzać przedziałów znajdujących się blisko początku:

$\texttt{BBBBBBC}\underline{\texttt{AACBBBBBBBBBBBBBBBBBBBB}}$