# Scrabble (Omówienie rozwiązań)

**XVIII OIJ, zawody drugiego stopnia**
2 marca 2024

Olimpiada Informatyczna Juniorów

Na początek zauważmy, że możemy zliczyć ile literek każdego typu jest w słowie $P$, którym dysponuje Bajtosia. Można stworzyć tablicę $ile[\cdot]$, która dla każdej pozycji w alfabecie zlicza powyższą wartość (w tym celu można skorzystać z konwersji typu `char` na `int` w C++ lub analogicznej funkcji `ord` w Pythonie).

## Rozwiązanie wolne

Możliwe jest rozciąganie okienka (przedziału) na słowie $S$. Wystarczy przetestować każdą możliwą pozycję rozpoczęcia $i$ i zakończenia $j$ obliczając tablicę $ile'[\cdot]$ dla fragmentu $S[i..j]$. Aby sprawdzić czy fragment znajdujący się w okienku od $i$ do $j$ może być skonstruowany z literek Bajtosi, wystarczy sprawdzić czy dla każdego znaku alfabetu $x$ zachodzi $ile'[x] \leq ile[x]$.

Tablicy $ile'[\cdot]$ nie trzeba obliczać za każdym razem od nowa. Jeżeli indeks $j$ przesunie się o jeden do przodu (poszerzenie okienka z prawej strony), wystarczy zwiększyć jedną komórkę tablicy. Każde poprawne okienko (w którym Bajtosia ma dostatecznie dużo literek każdego typu), które spełnia $j - i + 1 \geq \frac{|P|}{2}$ zwiększa wynik, który należy wypisać o 1.

Takie rozwiązanie nie otrzyma maksymalnej punktacji, ponieważ sprawdzanych par $(i, j)$ jest kwadratowo wiele względem długości wejścia, a czas sprawdzania pojedynczej pary jest proporcjonalny do rozmiaru alfabetu. Prowadzi to do uzyskania jedynie częściowej punktacji (około 46%).

Możliwe jest przyspieszenie czasu sprawdzania pojedynczego okienka do czasu stałego, jeżeli sprawdzać tylko przedłużenia dobrych okienek w tym miejscu tablicy $ile'$, które się zmieniło, jest to jednak nadal niewystarczające.

## Rozwiązanie wzorcowe

Okazuje się, że niewielka optymalizacja będzie prowadziła do uzyskania rozwiązania liniowego: sprawdzając w powyższym rozwiązaniu okienka zaczynające się na pozycji $i$, można znaleźć największe $j$, dla którego okienko $S[i..j]$ jest konstruowalne z dostępnych liter. Skoro więc okienko $S[i..j]$ jest dobre, to również okienko $S[i+1..j]$ będzie dobre: nie ma więc potrzeby sprawdzać każdej pary $(i, j)$, a w każdym kroku możliwe jest jedynie przesuwanie albo $i$ albo $j$ do przodu, łatwo też zaktualizować tablicę $ile'[\cdot]$ zmniejszając jedynie jedną komórkę. W ten sposób łączna liczba sprawdzonych kandydatów będzie $O(|S|)$ (jest liniowo wiele przesunięć indeksu $i$ oraz liniowo wiele przesunięć indeksu $j$).

Możliwe jest więc nawet osiągnięcie łącznego czasu obliczeń $O(|S|+|P|)$, co prowadziło oczywiście do uzyskania maksymalnej punktacji. Rozwiązania multiplikatywnie gorsze o stałą rozmiaru alfabetu miały szansę również na uzyskanie podobnych wyników, zależnie od jakości implementacji.



Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025".