

Fulle (szkic rozwiązania)

Autor zadania: **Karol Pokorski**



Najpierw można zliczyć ile kart każdej figury zostało podane na wejściu. Można do tego użyć słownika (`unordered_map` w C++ lub `dict` w Pythonie), sortowania lub naiwnie stworzyć tablicę par (figura, liczba jej wystąpień) przechodząc po elementach tablicy za każdym razem gdy wczytana zostanie kolejny znak wejścia (rozmiar wejścia to co najwyżej 52, więc będzie to dostatecznie szybkie).

W rozwiązaniu wystarczy jedynie wiedzieć ile jest figur, dla których Bajtek dysponuje dwoma, trzema i czterema kartami. Oznaczmy te liczby figur c_2 , c_3 oraz c_4 . Figury, dla których Bajtek ma dokładnie dwie karty mogą być użyte jedynie jako para w fullu.

Figury, dla których Bajtek ma trzy karty mogą być użyte jako para albo jako trójka w fullu. Spośród c_3 takich figur można założyć, że powstanie X trójek oraz $c_3 - X$ par.

Figury, dla których Bajtek ma cztery karty mogą być użyte jako co najwyżej dwie pary albo jako trójka w fullu. Spośród c_4 takich figur można założyć, że powstanie Y trójek oraz $2(c_4 - Y)$ par.

W ten sposób powstanie razem $c_2 + c_3 - X + c_4 - Y$ par oraz $X + Y$ trójek, a zatem $\min(c_2 + c_3 - X + c_4 - Y, X + Y)$ fulli. Możliwe jest teraz dalsze manipulowanie wzorami i próba jakiejś optymalizacji wyniku, ale ponieważ $c_2 + c_3 + c_4 \leq 13$, możliwe jest również naiwne sprawdzenie (podwójną pętlą) każdej sensownej pary (X, Y) i wybranie takiej, w której liczba powstałych fulli jest największa możliwa.

