Zbiór klocków (szkic rozwiązania)

Olimpiada Informatyczna Juniorów

Niech $C[i]$ oznacza liczbę wystąpień klocka z liczbą i. Wartości $C[\cdot]$ można obliczyć z użyciem struktury `unordered_map` w C++, `dict` w Pythonie lub z użyciem sortowania tablicy danej na wejściu. Jeszcze prościej, można użyć zwykłej tablicy zliczającej ograniczonej do indeksu N jeśli zauważymy, że dowolne optymalne rozwiązanie zawiera klocki z wartościami większymi niż N.

Zadanie można rozwiązać zachłannie lub z użyciem programowania dynamicznego. Rozwiązanie zachłanne sprowadza się do wybierania klocków w kolejności rosnących wartości liczb zapisanych na klockach, podobnie jak poniżej opisane rozwiązanie dynamiczne.

Niech $DP[i]$ oznacza największą liczbę klocków jakie można wybrać (zachowując warunek z zadania) z podzbioru klocków z wejścia o wartościach mniejszych lub równych i, czyli gdyby uznać klocki o wartościach większych niż i jako niedostępne.

Zakładamy, że $DP[0] = 0$ i obliczamy kolejne wartości $DP[1], DP[2], \ldots, DP[n]$. Obliczając wartość $DP[i]$ możemy przyjąć, że:

- nie bierzemy żadnego klocka z liczbą i i wtedy $DP[i] \leftarrow DP[i-1]$,
- albo bierzemy co najwyżej $i-1$ klocków z liczbami mniejszymi niż i i wszystkie klocki z liczbą i. Czyli $DP[i] \leftarrow \min(i-1, DP[i-1]) + C[i]$.

Dla każdego i wybieramy korzystniejszą alternatywę.

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