Informatyczny kącik olimpijski (113): Kieszonkowe

Zadanie (VIII Olimpiada Informatyczna Gimnazjalistów w roku szkolnym 2013/2014). *Danych jest n stosów, ponumerowanych od 1 do n. Każdy stos zawiera dokładnie dwie monety, ułożone jedna na drugiej. Ciąg $g_1, g_2, \ldots, g_n$ oznacza nominały monet, znajdujących się na górze kolejnych stosów, zaś ciąg $d_1, d_2, \ldots, d_n$ nominały monet, znajdujących się na dole kolejnych stosów. Staś, bohater zadania, może wykonać co najwyżej k ruchów. W każdym ruchu chłopiec wybiera dowolny niepusty stos i zabiera monetę ze szczytu tego stosu. Jaki jest największy możliwy zysk Stasia?*

Rozwiązanie dynamiczne

Zadanie możemy rozwiązać, korzystając z metody programowania dynamicznego. Niech $DP[i][j]$ oznacza maksymalny zysk dla problemu ograniczonego do stosów o numerach od 1 do i oraz maksymalnej liczbie ruchów równej j. Łatwo obliczyć wartości DP dla pierwszego stosu ($i = 1$):

- $DP[1][0] = 0$;
- $DP[1][1] = g_1$;
- $DP[1][j] = g_1 + d_1$, dla $j > 1$.

Przejdźmy teraz do obliczenia wyników dla większej liczby stosów. Dla każdego $i \in \{2, 3, \ldots, n\}$ oraz $j \in \{0, 1, \ldots, k\}$ aby obliczyć wartość $DP[i][j]$, należy rozpatrzyć trzy sytuacje:

- nie bierzemy ani jednej monety z i-tego stosu, wtedy wynikiem jest $DP[i-1][j]$;
- bierzemy jedną monetę z i-tego stosu, wtedy wynikiem jest $DP[i-1][j-1] + g_i$;
- bierzemy dwie monety z i-tego stosu, wtedy wynikiem jest $DP[i-1][j-2] + g_i + d_i$.

Wartością $DP[i][j]$ jest maksimum z trzech powyższych wartości. Oczywiście, druga sytuacja jest możliwa tylko dla $j > 0$, podobnie trzecia tylko dla $j > 1$. Największym zyskiem Stasia jest $DP[n][k]$.

Wszystkich stanów jest $O(nk)$. Obliczenie wyniku dla każdego stanu odbywa się w czasie stałym. Stąd złożoność czasowa rozwiązania wynosi $O(nk)$.

Rozwiązanie zachłanne

Okazuje się, że istnieje rozwiązanie o lepszej złożoności czasowej. Na początku podzielmy stosy na dwie grupy. Niech pierwsza grupa zawiera stosy, których górna moneta ma nominał nie mniejszy niż dolna moneta, nazwijmy je A-stosami. Pozostałe stosy niech tworzą drugą grupę i nazwijmy je B-stosami. Każdą z tych grup rozważmy oddzielnie.

Przypadek 1: A-stosy

Załóżmy, że mamy m A-stosów. Chcemy wybrać x monet, których sumaryczna wartość jest największa. Załóżmy dodatkowo, że w każdym ruchu możemy wybrać dowolną monetę (górną lub dolną). Nietrudno zauważyć, że aby zmaksymalizować zysk, należy wybrać x monet o największym nominale. Zatem posortujmy wszystkie monety malejąco według nominału. Otrzymujemy ciąg $a_1, a_2, \ldots, a_{2m}$, oznaczający wartości wszystkich monet. Największy zysk uzyskamy, jeśli weźmiemy następujące monety: $a_1, a_2, \ldots, a_k$.

Strategia zachłanna, która została opisana powyżej, jest również poprawna bez dodatkowego założenia o braniu dowolnej monety. Ustalamy zatem, że możemy brać tylko monety ze szczytu stosu. Przypuśćmy, że w optymalnym rozwiązaniu chcemy wziąć dwie monety a_i (górną) i a_j (dolną), które tworzą jeden stos. Skoro rozważamy A-stos ($a_i \geqslant a_j$) oraz ciąg a jest posortowany malejąco, to $i \leqslant j$. Zatem strategia zachłanna jest poprawna, ponieważ najpierw wybierze górną monetę a_i, a dopiero potem dolną a_j. Tutaj jest drobna subtelność, jeśli dwie monety mają ten sam nominał, wtedy pierwszeństwo ma górna.

Przypadek 2: B-stosy

Załóżmy, że mamy m B-stosów. Chcemy wybrać x monet, których sumaryczna wartość jest największa. Przypuśćmy, że w optymalnym rozwiązaniu istnieją dwa stosy (o numerach i oraz j), z których zabieramy po jednej monecie. Bez straty ogólności możemy założyć, że $g_i \leqslant g_j$. Wiadomo (z własności B-stosu), że $g_j < d_j$. Skoro $g_i \leqslant g_j$ oraz $g_j < d_j$, to $g_i < d_j$. Zatem wzięcie po jednej monecie ze stosów i i j nie jest optymalnym rozwiązaniem, gdyż bardziej opłaca się wziąć dwie monety z j-tego stosu. Stąd otrzymujemy, że w optymalnym rozwiązaniu nie ma dwóch stosów, z których zabieramy po jednej monecie. Innymi słowy, jest co najwyżej jeden stos, z którego zabieramy jedną monetę.

Zatem posortujmy stosy według malejącej sumy nominałów. Otrzymujemy ciąg $s_1, s_2, \ldots, s_n$, gdzie s_i oznacza sumę nominałów i-tego najcenniejszego stosu. Jeśli x jest parzyste, to wybieramy monety ze stosów $s_1, s_2, \ldots, s_{\frac{x}{2}}$. W przeciwnym przypadku rozważamy dwie sytuacje. Pierwsza z nich polega na wzięciu stosów $s_1, s_2, \ldots, s_{\frac{x}{2}}$ oraz monety o największym nominale spośród szczytów pozostałych stosów. Druga zaś polega na wzięciu stosów $s_1, s_2, \ldots, s_{\frac{x}{2}+1}$ bez monety o najmniejszym nominale spośród spodów wybranych stosów.

Podsumowanie

Rozwiązanie opiera się na rozpatrzeniu dla każdego $x \in \{0, 1, \ldots, k\}$ następującego przypadku: wybieramy x monet z A-stosów oraz $k - x$ monet z B-stosów. Wynikiem jest maksimum spośród wyników dla tych przypadków. Całe rozwiązanie działa w czasie $O(n \cdot \log(n))$. Szczegóły implementacyjne pozostawiam Czytelnikowi jako ćwiczenie.

Bartosz ŁUKASIEWICZ