Labirynty (szkic rozwiązania)

Autor zadania: **Karol Pokorski**

Olimpiada Informatyczna Juniorów

Zadanie można próbować rozwiązać rozważając kolejne opisy poziomów zgodnie z kolejnością na wejściu. Można policzyć liczbę diamentów po ukończeniu danego poziomu (niech będzie to poziom numer i), gdyby założyć, że wcześniej gracz nie wykona żadnej operacji zamiany (jest to po prostu suma elementów ciągu A do i-tej pozycji). Jeżeli po żadnym poziomie liczba diamentów nie spada poniżej 0 to na wyjściu należy wypisać 0.

Jeżeli jednak poziom numer i jest pierwszym poziomem, na którym balans diamentów byłby ujemny, konieczne jest wykonanie pewnej zamiany w dotychczas wczytanym fragmencie ciągu. Jedną z możliwości jest oczywiście wykonać zamianę potwora na końcu poziomu i. Pozwala to jedną operacją przywrócić dodatni balans diamentów, a więc pokazuje, że jedna operacja wystarczy. Czasami jednak jedną operacją można zdziałać więcej: opłaca się bowiem wykonać zamianę w dowolnym poziomie nie późniejszym niż i-ty, w którym był potwór wymagający najdroższego miecza. Przyniesie to więcej dodatkowych diamentów w skrzyni, a więc najwyższy możliwy balans również na koniec i-tego poziomu. Należy więc zaktualizować odpowiednio balans i odpowiednio podmienić ciąg wejściowy aby dwa razy nie zamienić tego samego potwora w skrzynię z diamentami.

Naiwne znajdowanie potwora wymagającego najdroższego miecza prowadziłoby do algorytmu o złożoności $O(N^2)$, która w tym zadaniu była nieakceptowalna (rozwiązania o tej złożoności powinny być ocenione na ok. 65% punktów). Możliwe jest jednak przechowywanie dotychczas napotkanych potworów na kolejce priorytetowej (wykorzystując jedną ze standardowych struktur `std::priority_queue` lub `std::set` lub `std::multiset` w C++ lub `heapq` w Pythonie). Wówczas znalezienie optymalnego poziomu do wykonania zamiany będzie każdorazowo kosztowało czas $O(\log N)$, a usunięcie tego potwora również zajmie czas $O(\log N)$, co pozwoli uzyskać satysfakcjonującą złożoność $O(N \log N)$ całego algorytmu.

W poprawnej drabinie każda igła połączona jest bezpośrednio z jedną lub trzema innymi igłami. Można więc utrzymywać tablicę indeksowaną numerami igieł. W i-tej komórce tej tablicy może być tablica trzyelementowa zawierająca numery igieł sąsiadujących z igłą numer i. Wczytując połączenie między igłami u_i oraz v_i należy więc dopisać v_i jako element podtablicy igły u_i oraz analogicznie należy dopisać u_i jako element podtablicy igły v_i. Jeżeli podczas wczytywania okazało się, że jakaś igła ma więcej niż trzy połączenia z innymi igłami, należy wypisać `NIE` i zakończyć program.

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”.