Ciężarówki II (szkic rozwiązania)

Autor zadania: **Mateusz Radecki**
Opracowanie: **Mateusz Olszewski, Jacek Salata**
Opis rozwiązania: **Karol Pokorski**

Olimpiada Informatyczna Juniorów

Nietrudno domyślić się, że mamy do czynienia z zadaniem grafowym. Miasta Bajtocji stanowią wierzchołki grafu, a drogi je łączące są krawędziami z wagą równą kosztowi przejazdu tą drogą.

Nieco nietypowym jest fakt, że wagą ścieżki między wierzchołkami w grafie jest waga najcięższej krawędzi występującej na ścieżce (zazwyczaj jest to suma wag), uniemożliwia to stosowanie standardowych algorytmów dla problemu najkrótszych ścieżek (np. algorytmu Dijkstry). Z drugiej jednak strony, możemy zawuażyć, że aby sprawdzić czy dwa wierzchołki można połączyć tylko krawędziami o koszcie nie przekraczającym z góry założonego limitu, nie musimy nawet znać dokładnej struktury grafu. Wystarczy wiedzieć czy wierzchołki są w tej samej czy różnej spójnej składowej podgrafu zawierającego krawędzie o koszcie co najwyżej równym założonemu limitowi. Nie jest przy tym istotne jakimi konkretnie krawędziami przejdziemy.

Możemy więc zacząć od grafu, który nie zawiera żadnych krawędzi i dokładać je w kolejności rosnących kosztów. Utrzymujemy strukturę Union-Find, w której wierzchołki leżące w tej samej spójnej grafu są w tym samym zbiorze, a wierzchołki leżące w różnych spójnych są w różnych zbiorach. Dodanie krawędzi powoduje połączenie odpowiednich zbiorów. Przy odpowiedniej implementacji (kompresja ścieżek lub łączenie według rang) można pokazać, że zamortyzowany koszt takiej operacji nie przekracza $O(\log N)$. W dowolnym momencie wykonania algorytmu, dla każdego zbioru w strukturze Union-Find utrzymujemy dodatkowo liczbę wierzchołków a_i z tego zbioru, w których początkowo znajdowały się ciężarówki oraz liczbę wierzchołków b_i z tego zbioru, w których docelowo mają się znaleźć ciężarówki. Możemy, więc w danym momencie przenieść jedynie $\min(a_i, b_i)$ ciężarówek na swoje miejsce i oczywiście opłaca się to zrobić jak najwcześniej (aby jak najczęściej użyć krawędzi o jak najmniejszych kosztach). A zatem, w momencie dodania krawędzi grafu, gdy pewne dwie spójne o parametrach (a', b') oraz (a'', b'') są łączone, uzyskujemy nową spójną o parametrach $(a' + a'', b' + b'')$ i dokładnie $\min(a' + a'', b' + b'') - \min(a', b') - \min(a'', b'')$ nowych ciężarówek przejedzie dodawaną krawędzią grafu jako najdroższą na swojej ścieżce. Dodajemy tę liczbę razy wagę krawędzi do wyniku.

Ostatecznie, zakładając, że $M = \Omega(N)$, łączny czas działania algorytmu jest zdominowany przez złożoność sortowania krawędzi i wynosi $O(M \log M)$.

Niektóre inne możliwe rozwiązania

- spowolnione warianty rozwiązania wzorcowego,
- próbowanie wszystkich $k!$ możliwych dopasowań wierzchołków źródłowych do docelowych,
- rozwiązania dla drzew lub ogólnie: na przykład znajdowanie najtańszej trasy od jeszcze nieprzeniesionych ciężarówek do wszystkich możliwych pozycji docelowych (dla drzew można to robić dowolnym algorytmem przeszukiwania grafu) i zachłanne dopasowanie do siebie końców najtańszej ścieżki (usuwając znaleziony wierzchołek źródłowy i docelowy z rozważań),
- bardziej skomplikowane i wolniejsze od wzorcowego rozwiązanie oparte o min-cost max-flow.

Olimpiada finansowana jest ze środków Ministerstwa Edukacji i Nauki w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".