Zadanie 4, Lista 3

Kuba Nowak

17 kwietnia 2020

W tym zadaniu mamy znaleźć liczbę par wierzchołków odległych od siebie o C w zadanym drzewie T. W tym celu zastosujemy algorytm oparty na zasadzie dziel i zwyciężaj. Będzie on obliczał liczbę wierzchołków odległych o C w drzewie i zwracał odległość wierzchołka spajającego od każdego innego wierzchołka w drzewie.

Dzielenie

W pierwszym kroku staramy się podzielić drzewo w miarę możliwości na części o równych rozmiarach. Robimy to wyszukując centroid w czasie liniowym do wielkości drzewa.

Mając znaleziony centroid rozbijamy drzewo na poddrzewa wychodzące z centroidu i dla każdego z poddrzew zapamiętujemy wierzchołek, który łączy się z centroidem, będzie on wierzchołkiem spajającym dla tego poddrzewa. Następnie wywołujemy się rekurencyjnie na poddrzewie.

Scalanie

Aby algorytm oparty na zasadzie dziel i zwyciężaj miał sens, scalenie musi nastąpić w czasie liniowym. Po wywołaniach rekurencyjnych na poddrzewach mamy wartości $X_1, X_2, ..., X_k$ oraz wektory $w_1, w_2, ...w_k$, dla k będącego liczbą powstałych poddrzew, gdzie X_i oznacza liczbę wierzchołków odległych o C w danym poddrzewie, a w_i jest wektorem odległości wierzchołka spajającego i-tego poddrzewa od każdego innego wierzchołka w tym poddrzewie.

Musimy teraz znaleźć liczbę B, będącą liczbą wierzchołków odległych o C w naszym drzewie. Będzie to suma wierzchołków odległych o C w każdym z poddrzew, liczby wierzchołków odległych o C z różnych poddrzew (oznaczmy ją przez Y) oraz liczby wierzchołków odległych o C od centroidu.

W celu policzenia Y, dla każdej odległości d mniejszej równej C, zliczmy liczbę wierzchołków odległych o d od centroidu. Niech będzie to tablica D (w rzeczywistości będziemy chcieli zapewne przechowywać to jako zbiór par, aby operacje były liniowe względem n, a nie C, dla uproszczenia przyjmijmy jednak, że jest to tablica). Podobne tablice wyliczmy dla każdego z poddrzew. Niech tablica D_i będzie tablicą dla i-tego poddrzewa.

Mając powyższe tablice pomocnicze można zacząć liczyć Y. W tym celu inicjalizujemy Y na zero i przechodzimy po każdym wektorze w_i. Dla każdego v będącego j-tym elementem z w_i (czyli j-tym wierzchołkiem z i-tego poddrzewa) liczymy wartość Z będącą liczbą wierzchołków odległych o C od wierzchołka v, w innych poddrzewach.

$$Z = D[C - (w_i[j] + c_i)] - D_i[C - (w_i[j] + 2c_i)]$$

gdzie c_i jest wagą krawędzi z wierzchołka spajającego i do centroidu. W celu obliczenia Z wartości, liczymy odległość v od centroidu i następnie sprawdzamy ile wierzchołków jest takich, że można osiągnąć między nimi, a wierzchołkiem v, drogę długości C przechodzącą przez centroid. Jedak powoduje to, że mogą powstać drogi z v do i-tego poddrzewa, czyli drogi wewnątrz jednego poddrzewa przechodzące przez centroid, dlatego musimy odjąć takie drogi od ogólnej liczby dróg i w tym celu odejmujemy wierzchołki z poddrzewa i, które leżą w odległości $C - (w_i[j] + 2c_i)$ od wierzchołka spajającego poddrzewa i. Ponieważ takie wierzchołki dadzą z v drogę długości C, przechodzącą przez centroid.

Po wyliczeniu wartości Z dodajemy ją do Y. To spowoduje, że na koniec przechodzenia po wektorach w będziemy mieli w Y liczbę dróg między wierzchołkami z dwóch różnych poddrzew o długości C. Więc w celu obliczenia liczby wszystkich takich dróg w drzewie musimy przeprowadzić następujące sumowanie:

$$B = D[C] + Y + \sum_i X_i$$

Czyli do wierzchołków odległych o C od centroidu dodajemy wierzchołki odległe o C z dwóch różnych poddrzew i wierzchołki odległe o C z tych samych poddrzew.

Mając wyliczoną liczbę par odległych o C w drzewie, należy wyliczyć jeszcze odległość każdego wierzchołka od wierzchołka spajającego całego drzewa, co można zrobić w czasie liniowym przy pomocy DFS-a.

Złożoność

Zauważmy, że scalanie wykonujemy w czasie liniowym od liczby wierzchołków w drzewie (n), ponieważ:

- Zbiór tablic D_i wyznaczamy w $O(n)$
- Tablicę D wyznaczamy w $O(n)$
- Y wyznaczamy przechodząc wszystkie wektory w, a mają one łącznie $n-1$ elementów, czyli czas jest $O(n)$
- B wyznaczamy w czasie $O(n)$
- Odległości od wierzchołka spajającego grafu wyznaczamy w $O(n)$

Algorytm będzie działał w czasie $O(n \log n)$, ponieważ w każdym kroku algorytmu redukujemy rozmiar problemu co najmniej o połowę. Głębokość drzewa wywołań jest więc co najwyżej $\log n$, a na każdym poziomie przeprowadzamy $O(n)$ operacji.

Szukanie centroidu w czasie liniowym

Ważną częścią powyższego algorytmu jest wyszukanie centroidu w czasie liniowym. Można to osiągnąć przy pomocy DFS-a. Najpierw uruchamiamy DFS w dowolnym wierzchołku i przy jego pomocy dla każdego wierzchołka liczymy wielkość jego poddrzewa. Następnie zaczynając z wierzchołka w którym wywołaliśmy DFS-a przechodzimy po drzewie tak, że za każdym razem idziemy do wierzchołka, którego poddrzewo jest większe niż $n/2$, idziemy tak długo, aż nie możemy przejść do następnego wierzchołka (nigdy nie wracamy się), ostatni odwiedzony wierzchołek jest centroidem.