

Lista 7

Zadanie 1

1 Zadanie

Jaki będzie czas działania procedury BFS, jeśli graf wejściowy jest reprezentowany przez macierz sąsiedztwa, a algorytm jest zmodyfikowany w taki sposób, żeby działał poprawnie dla tej reprezentacji?

2 Rozwiązanie

Mamy graf $G = (V, E)$ gdzie V to zbiór wierzchołków, a E to zbiór krawędzi. Załóżmy, że $|V| = n$. Reprezentujemy G przy pomocy macierzy sąsiedztwa, gdzie indeksy kolumn i wierszy odpowiadają indeksom wierzchołków V . Czyli dla każdej krawędzi $(u, v) \in E$ mamy $G_M[\text{index}(u)][\text{index}(v)] = 1$, a pozostałe wartości tej macierzy to zera. Oczywiście musimy określić funkcję $\text{index} : V \xrightarrow[n-1]{na} [0, n]$ określającą reprezentację wierzchołka jako unikalnej liczby naturalnej.

Procedura BFS na początku przyporządkowuje odległości od wierzchołka startowego s do wszystkich wierzchołków w grafie jako ∞ (oprócz s któremu daje 0). Niniejsze odległości algorytm może zapisywać np. do tablicy „dist” długości n z zachowaniem indeksów wierzchołków z macierzy G_M . W ten sam sposób mogą zostać przechowywane wskaźniki na „poprzednie” wierzchołki, taką tablicę nazwiemy „prev” przy czym $\text{prev}[\text{index}(s)] = \text{index}(s)$.

W tej zmodyfikowanej procedurze BFS kolejka Q oczywiście przechowywałaby indeksy aktualnie rozważanych wierzchołków. Wówczas inicjacja tej kolejki również podlega zmianie: $Q \leftarrow [\text{index}(s)]$.

W środku pętli która działa dopóki Q nie jest pusta zawarta jest następna pętla **for** która przegląda wszystkie krawędzie z aktualnie rozważanego wierzchołka u do wierzchołków incydentalnych z nim. Niniejsza pętla musi rozważyć cały wiersz $G_M[\text{index}(u)]$. Algorytm przegląda cały wiersz w poszukiwaniu 1, które oznaczają krawędź między u a innym wierzchołkiem. Czyli przegląda listę długości n . Co daje nam złożoność obliczeniową równą $O(n)$. Zakładając, że graf G jest spójny to główna pętla, która dzieli graf na „warstwy”, również wykona się n razy. Co daje nam całkowitą złożoność algorytmu równą $O(n^2)$, jako że wcześniej wspomniana pętla **for** jest zagnieżdżona w pętli **while** $|Q| > 0$.