Algorytm Prima

Dziś będziemy też implementować algorytm Prima. Jest to algorytm zachłanny, który wyznacza minimalne drzewo rozpinające graf (czyli robi to samo co algorytm Kruskala). W algorytmie tym musimy podać wierzchołek początkowy. Zakładamy, że rozważamy graf spójny i nieskierowany.

Szkic algorytmu:

1) Tworzymy listę „odwiedzonych" wierzchołków. Dodajemy do niej wierzchołek początkowy.
2) Tworzymy kolejkę priorytetową Q do przechowywania krawędzi, początkowo jest ona pusta. Priorytetem będzie waga krawędzi (w kolejce preferujemy niskie wagi)
3) Tworzymy listę krawędzi (początkowo pustą), która będzie zawierać wyniki (tzn. krawędzie wchodzące w skład drzewa MST).
4) Początkowo, do kolejki priorytetowej Q dodajemy wszystkie krawędzie wychodzące z wierzchołka startowego

Dopóki liczba odwiedzonych wierzchołków jest mniejsza niż liczba wierzchołków w grafie robimy tak:

1) Zdejmujemy krawędź z przodu kolejki priorytetowej (czyli tą o najniższej wadze).
2) Jeśli zdjęta krawędź prowadzi do wierzchołka już odwiedzonego (sprawdzamy na liście odwiedzonych wierzchołków) to nie robimy nic i wracamy do punktu 1, jeśli jednak krawędź prowadzi do nieodwiedzonego wierzchołka (o indeksie Z) to wykonujemy co następuje:
   a) Dodajemy ten wierzchołek (o indeksie Z) do listy odwiedzonych wierzchołków
   b) Dodajemy krawędź do listy krawędzi wchodzących w skład MST.
   c) Do kolejki priorytetowej Q dodajemy wszystkie krawędzie wychodzące z wierzchołka Z (można doprecyzować, że chodzi o dodanie wszystkich krawędzi wychodzących z wierzchołka Z, które prowadzą do nieodwiedzonych wierzchołków, choć tak naprawdę nie ma to znaczenia, bo krawędzie prowadzące do wierzchołków odwiedzonych będą bez procedowana „ściągane" z kolejki). Oczywiście mówimy tu o kolejce priorytetowej, więc ona automatycznie uaktualni kolejność swoich „składników" by uwzględnić wagi nowo dodanych krawędzi 😊

Moje uwagi:

1) Rezultat algorytmu – rozumiany jako suma krawędzi w MST, nie będzie zależał od wyboru wierzchołka początkowego, kolejność dodawania wierzchołków (i krawędzi) do list (a nawet sam wybór krawędzi) będzie jednak zależeć od wierzchołka początkowego.
2) Algorytm powinien dawać takie same wyniki jak algorytm Kruskala (zbiór krawędzi wchodzących w skład MST a w szczególności suma wag krawędzi wchodzących w skład MST). Myślę, że zgodność sum wag (z algorytmem Kruskala) warto sprawdzić, bo ułatwia to wyszukanie błędów.
3) Jak już o błędach w implementacji mowa, to przypominam, że mówimy tu o grafie nieskierowanym. Jeśli krawędź (x,y, o wadze z) łączy wierzchołki x i y to działa ona w „dwie strony". Gdy wczytujemy linijkę z pliku ze spisem krawędzi to dodając do listy krawędzi grafu np. krawędź (1,2 o wadze 4) trzeba też dodać krawędź (2,1 o wadze 4). Jest to ważne, bo dotychczas mówiliśmy często o grafach skierowanych. Również algorytm Kruskala nie jest czuły na „pominięcie" dodania do listy krawędzi krawędzi „powrotnej", ale algorytm Prima już jest w tym zakresie czuły.
   Dla ułatwienia do treści zadania dodam Państwu dwie listy krawędzi:
   -jedną bez jawnie podanych krawędzi powrotnych

-drugą (opisującą ten sam graf) z jawnie wypisanymi krawędziami powrotnymi**. Generalnie jak skorzystają Państwo z pliku z jawnie podanymi krawędziami powrotnymi to można zapomnieć o całym punkcie 3 uwag (będzie on zrealizowany automatycznie 😊).**