Algorytmy do zestawu nr 3[1]

1. Algorytmy pomocnicze

1.1. Nadanie wartości początkowych atrybutów d oraz p dla wierzchołków grafu G

Init(G, s) *// G - graf wejściowy, s - wybrany wierzchołek-źródło*

1. `for` każdy wierzchołek v należący do grafu G
2. `do` $d_s(v) = \infty$
3. $p_s(v) = $ Nil
4. $d_s(s) = 0$

1.2. Relaksacja krawędzi (u, v)

Relax(u, v, w) *// u, v - wierzchołki połączone krawędzią (u, v) poddaną relaksacji; w - wagi*

1. `if` $d_s(v) > d_s(u) + w(u, v)$
2. `then` $d_s(v) = d_s(u) + w(u, v)$
3. $p_s(v) = u$

---

2. Algorytm Dijkstry

Dijkstra(G, w, s)

1. Init(G, s)
2. $S = \varnothing$ *// S - zbiór "gotowych" wierzchołków; na początku jest pusty*
3. `while` $S \neq$ zbiór wszystkich wierzchołków G
4. `do` $u =$ wierzchołek o najmniejszym $d_s(u)$ spośród niegotowych wierzchołków: $u \notin S$
5. $S = S \cup u$
6. `for` każdy wierzchołek $v \notin S$ będący sąsiadem u
7. `do` Relax(u, v, w)

---

[1] Na podstawie: Cormen Thomas H., Leiserson Charles E., Rivest Ronald L., *Wprowadzenie do algorytmów*, Wyd. 4, Warszawa, Wydawnictwo Naukowo - Techniczne, 2001, ISBN 83-204-2665-0.

3. Minimalne drzewo rozpinające[2]

- G - graf wejściowy,
- n - liczba wierzchołków grafu G,
- (u, v) - krawędź z wierzchołka u do v,
- T - minimalne drzewo rozpinające.

3.1. Algorytm Prima

Początkowo wierzchołki grafu dzielimy na dwie części: dowolny startowy wierzchołek dodajemy do pustego drzewa T, pozostałe wierzchołki tworzą zbiór W. W każdym kroku **analizujemy tylko krawędzie łączące T z W:**

1. Wybieramy krawędź (u, v) o najmniejszej wadze (tzw. krawędź lekka).
2. Dodajemy (u, v) do drzewa T, jednocześnie usuwając odpowiedni wierzchołek z W.

Pętlę wykonujemy póki T nie zawiera wszystkich n wierzchołków grafu wejściowego.

3.2. Algorytm Kruskala

Wiadomo, że do drzewa T będą należały wszystkie wierzchołki. Początkowo nie są połączone (należą do różnych drzew). Sortujemy krawędzie grafu G według niemalejących wag. Dla każdej krawędzi (u, v) od najmniejszej wagi: jeżeli dodanie krawędzi (u, v) nie spowoduje powstania cyklu (czyli: krawędź (u, v) łączy różne drzewa), to dodajemy ją do T.
Algorytm można zakończyć przed końcem pętli, o ile T zawiera już $n - 1$ krawędzi.

[2]Na podstawie: V. K. Balakrishnan, *Schaum's Outline of Theory and Problems of Graph Theory*, McGraw-Hill Education - Europe, 1997, ISBN 0-07-005489-4.