Algorytm Kruskala

Jakub Orchowski, Jakub Szych

# Opis Algorytmu

**Algorytm Kruskala** to popularny, **zachłanny** algorytm służący do znajdowania **Minimalnego Drzewa Rozpinającego** (ang. *Minimum Spanning Tree*, MST) w spójnym, nieskierowanym i ważonym grafie. MST to podgraf, który łączy wszystkie wierzchołki grafu za pomocą krawędzi o najmniejszej możliwej sumie wag, nie tworząc przy tym żadnych cykli.

Działanie algorytmu opiera się na prostej zasadzie:

1. Utwórz listę wszystkich krawędzi w grafie.
2. Posortuj krawędzie w porządku **niemalejącym** według ich wag.
3. Przechodź przez posortowaną listę krawędzi. Dla każdej krawędzi sprawdzaj, czy jej dodanie do już wybranych krawędzi **nie spowoduje powstania cyklu**.
4. Jeśli dodanie krawędzi nie tworzy cyklu, dodaj ją do zbioru krawędzi tworzących MST.
5. Kontynuuj proces, aż do MST zostanie dodane V−1 krawędzi, gdzie V to liczba wierzchołków w grafie.

Do efektywnego wykrywania cykli najczęściej używa się struktury danych zwanej **zbiorem rozłącznym** (ang. *Disjoint Set Union*, DSU).

# Złożoność Obliczeniowa

Złożoność czasowa algorytmu Kruskala zależy od dwóch głównych operacji: sortowania krawędzi i operacji na strukturze DSU.

* **Sortowanie krawędzi**: Jeśli graf ma E krawędzi, złożoność sortowania wynosi O(ElogE).
* **Operacje na DSU**: Wykonujemy E operacji find i co najwyżej V−1 operacji union. Dzięki optymalizacjom (unia według rangi i kompresja ścieżki), zamortyzowany czas pojedynczej operacji na DSU jest niemal stały, bliski O(α(V)), gdzie α to odwrotność funkcji Ackermanna, która rośnie ekstremalnie wolno.

Dominującym czynnikiem jest sortowanie, więc całkowita złożoność obliczeniowa algorytmu Kruskala wynosi **O(ElogE)** lub równoważnie **O(ElogV)**, ponieważ w grafie spójnym liczba krawędzi E jest co najmniej V−1.

**Sposób Wprowadzania Danych**

Aby program mógł przetworzyć graf, dane muszą być wprowadzone w określonym formacie z klawiatury:

1. **Liczba wierzchołków i krawędzi**: Najpierw program pyta o dwie liczby całkowite:

* Liczbę wierzchołków (V).
* Liczbę krawędzi (E).

1. **Definicja krawędzi**: Następnie program prosi o wprowadzenie E krawędzi, każdej w osobnej linii. Wierzchołki są numerowane od 0 do V-1. Każdą krawędź należy zdefiniować przez podanie trzech liczb całkowitych oddzielonych spacjami:

* u: numer wierzchołka początkowego.
* v: numer wierzchołka końcowego.
* w: waga krawędzi łączącej u i v.

**Przykład wprowadzania danych:**

Podaj liczbę wierzchołków w grafie: 4

Podaj liczbę krawędzi w grafie: 5

Podaj dane dla każdej krawędzi w formacie 'u v w'

(wierzchołek\_początkowy wierzchołek\_końcowy waga)

Wierzchołki numeruj od 0 do 3.

Krawędź 1: 0 1 10

Krawędź 2: 0 2 6

Krawędź 3: 0 3 5

Krawędź 4: 1 3 15

Krawędź 5: 2 3 4

**Przykład Działania Algorytmu**

Rozważmy graf z powyższego przykładu z 4 wierzchołkami i 5 krawędziami.

1. **Lista krawędzi i ich wag**:

* (2, 3, 4)
* (0, 3, 5)
* (0, 2, 6)
* (0, 1, 10)
* (1, 3, 15)

1. **Sortowanie krawędzi**: Krawędzie są sortowane rosnąco według wag.
2. **Wynik**: Algorytm kończy działanie, ponieważ wybrano V−1=3 krawędzie.

* **Minimalne Drzewo Rozpinające** składa się z krawędzi: (2, 3), (0, 3), (0, 1).
* **Całkowita waga MST** = 4+5+10=19.

**Obsługa błędów**  
Program sprawdza poprawność danych wejściowych:

* Weryfikuje, czy numery wierzchołków mieszczą się w dozwolonym zakresie.
* Sprawdza, czy dane są wprowadzone w odpowiednim formacie (trzy liczby oddzielone spacją).
* Informuje użytkownika o błędach i pozwala poprawić dane.
* Program nie zgłasza błędu w przypadku podania tej samej krawędzi kilka razy. Algorytm Kruskala działa poprawnie nawet w takiej sytuacji – nadmiarowe krawędzie są ignorowane, jeśli ich dodanie powodowałoby cykl.

**Podsumowanie**

Algorytm Kruskala jest jednym z najprostszych i najwydajniejszych sposobów znajdowania minimalnego drzewa rozpinającego w grafie. Dzięki zastosowaniu struktury zbiorów rozłącznych oraz sortowaniu krawędzi, zapewnia wysoką efektywność i poprawność działania nawet dla dużych grafów. Program umożliwia łatwe wprowadzanie danych oraz informuje użytkownika o ewentualnych błędach wejściowych, co czyni go intuicyjnym w obsłudze.