

SK2. Wizualizacja algorytmów grafowych II

Dokumentacja wstępna

Michał Kielak, Michał Uziak

16 grudnia 2013

1 Cel projektu

Celem projektu jest implementacja oprogramowania do wizualizacji wybranych algorytmów kolorowania grafu.

2 Algorytm

W projekcie zostanie użyty algorytm Kruskala, wyznaczający minimalne drzewo rozpinające dla spójnego, ważonego grafu nieskierowanego. Algorytm został stworzony przez Josepha Kruskala w 1956 roku.

2.1 Opis działania algorytmu Kruskala

Kroki tworzenia drzewa w algorytmie Kruskala:

- utworzenie lasu z wierzchołków (każdy wierzchołek jest osobnym drzewem) - zbiór L
- posortowanie krawędzi wg wag - zbiór S
- dopóki $S > 0$
 - wybranie i usunięcie krawędzi o najmniejszej wadze ze zbioru S
 - jeśli krawędź łączyła dwa różne drzewa, dodanie jej do lasu L
 - jeśli krawędź nie łączyła dwóch różnych drzew, usunięcie

2.2 Złożoność obliczeniowa

Poszczególne fazy algorytmu mają złożoności równe:

- sortowanie krawędzi według wag - złożoność $O(E \log V)$
- tworzenie drzewa - $O(E \alpha(E, V))$

Zatem całkowita złożoność algorytmu wynosi $O(E \log V)$

gdzie:

E - liczba punktów

V - liczba krawędzi

α - odwrotność funkcji Ackermanna

3 Implementacja

Projekt zostanie wykonany w języku Python, przy użyciu biblioteki Qt4. Założeniem programu jest działanie w czasie rzeczywistym - użytkownik wprowadza przykładowy graf, następnie przechodząc przez kolejne etapy może prześledzić kroki tworzenia minimalnego drzewa rozpinającego. Algorytm kończy działanie, gdy wyznaczy minimalne drzewo rozpinające. Przejście między kolejnymi krokami budowania drzewa będzie dla użytkownika niezauważalny, dlatego czas wykonania algorytmu może zostać pominięty.

3.1 Struktury danych

Główną strukturą danych wykorzystywaną w projekcie będzie Graph z modułu networkx przeznaczonego dla grafów i sieci.

3.2 Format wejścia i wyjścia

Użytkownik będzie miał możliwość wprowadzenia danych na dwa sposoby:

- przez graficzny interfejs użytkownika (rysowanie punktów, krawędzi, podawanie wag)
- przez wczytanie plików tekstowych

Format pliku wejściowego:

```
MACIERZ WAG
1 1 1 0
3 4 2 1
WSPÓŁRZĘDNE WIERZCHOŁKÓW
x y z
x y z
```

Po wykonaniu programu, użytkownik będzie miał możliwość zapisania otrzymanego grafu w postaci pliku graficznego.

3.3 Sytuacje wyjątkowe

Możliwe sytuacje wyjątkowe:

- wczytanie źle sformatowanego pliku - pojawienie się okna z informacją o błędzie
- połączenie dwóch tych samych wierzchołków więcej niż jedną krawędzią - zignorowane, wybranie pierwszej ze zdefiniowanych krawędzie
- osierocone wierzchołki - usunięcie zbędnych wierzchołków

3.4 Dodatkowe opcje

- kolorowanie krawędzi
- ustawianie wielkości punktów
- zapisywanie, jako pliki graficzne, kolejnych kroków tworzenia drzewa