Minimalne drzewa rozpinające

Weronika Siwek, Marek Biedrzycki

KNM UŚ

26-28 listopada 2010

Ostrzeżenie

Niektóre pojęcia pojawiające się podczas tego referatu są naszymi autorskimi tłumaczeniami z języka angielskiego. Nie udało nam się znaleźć ich polskich odpowiedników. Dlatego nazwy, które Szanowny Słuchacz może znaleźć w literaturze, **mogą różnić się** od użytych w tej prezentacji.

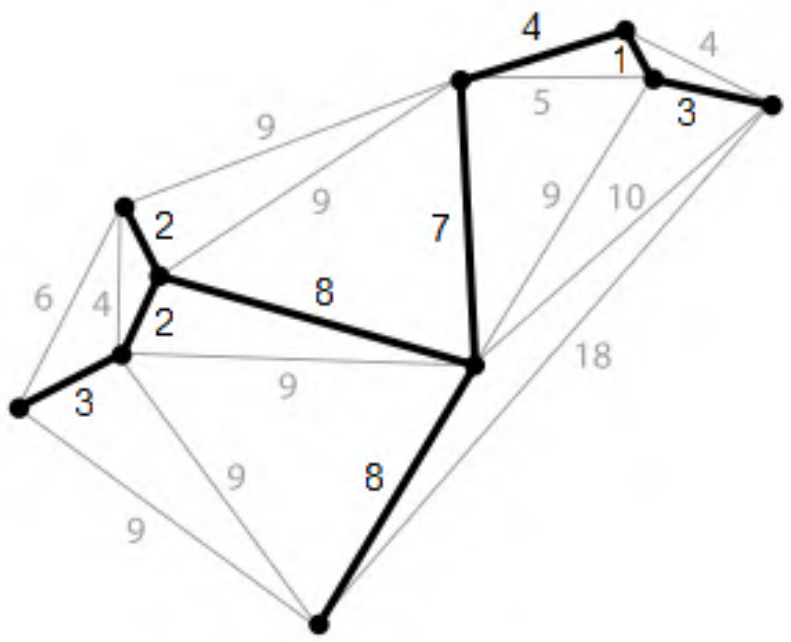
Zapraszamy do Utrechtu. Budynki uniwersytetu w tym mieście połączone są na wysokości pierwszego piętra korytarzami takimi jak ten:

Wyobraźmy sobie firmę mieszczącą się w kilkunastu (kilkudziesięciu, kilkuset ☺) budynkach. Firma ta chce wybudować podobne korytarze pomiędzy swoimi budynkami. Znany jest koszt budowy każdego możliwego korytarza. Ponadto, szefowi firmy zależy, by:

1. Między każdymi dwoma budynkami istniało (niekoniecznie bezpośrednie) połącznie;
2. Łączny koszt budowy korytarzy był (a jakże) jak najniższy.

Czy potrafisz rozwiązać powstały problem?

Problem można zobrazować za pomocą grafu, w którym wierzchołki odpowiadają poszczególnym budynkom, a krawędzie możliwym do wybudowania korytarzom. Każdej krawędzi przypisana jest liczba oznaczająca koszt budowy danego korytarza. Zadanie sprowadza się do znalezienia tzw. minimalnego drzewa rozpinającego:



Czym będziemy się zajmować?

Rozważmy spójny, nieskierowany graf z wagami $G = (V, E, c)$, gdzie:

- V jest zbiorem wierzchołków, $|V| = n$;
- E jest zbiorem krawędzi, $E \subseteq \{\{v, w\} \colon v, w \in V\}$, $|E| = m$;
- c_e oznacza wagę krawędzi $e = \{v, w\} \in E$.

Drzewo rozpinające graf

Drzewo rozpinające

$T \subseteq E$ nazywamy **drzewem rozpinającym** graf $G = (V, E)$ wtedy i tylko wtedy, gdy graf (V, T) jest spójny i nie zawiera cykli. Drzewo rozpinające graf zapisywać będziemy w skrócie ST (*Spanning Tree*).

ST - warunki równoważne [Wilson, 2008]

Następujące warunki są równoważne:

(i) $T \subseteq E$ jest drzewem rozpinającym graf $G = (V, E)$;

(ii) graf (V, T) jest spójny i $|T| = |V| - 1$;

(iii) graf (V, T) nie zawiera cykli i $|T| = |V| - 1$;

(iv) w grafie (V, T) dla dowolnych wierzchołków $v, w \in V$ istnieje jedyna ścieżka o końcach v, w.

Minimum Spanning Tree

Dla danego grafu $G = (V, E, c)$ spróbujmy znaleźć drzewo rozpinające T. Wagą drzewa T nazywamy sumę wag krawędzi tego drzewa:

$$\sum_{e \in T} c_e$$

Nasze zadanie - MST problem:

Znaleźć drzewo rozpinające grafu $G = (V, E, c)$ o najmniejszej z możliwych wag.

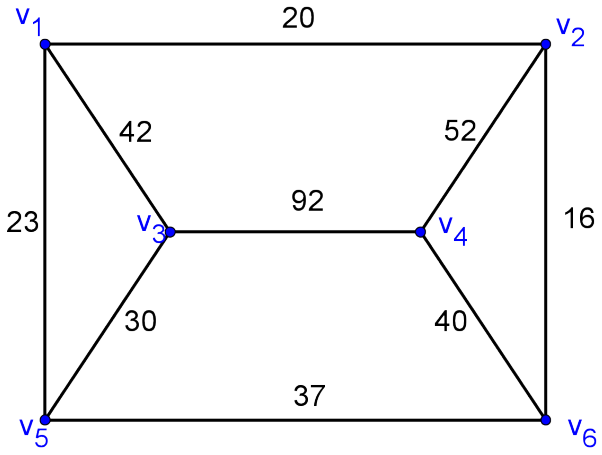
Rozcięcia

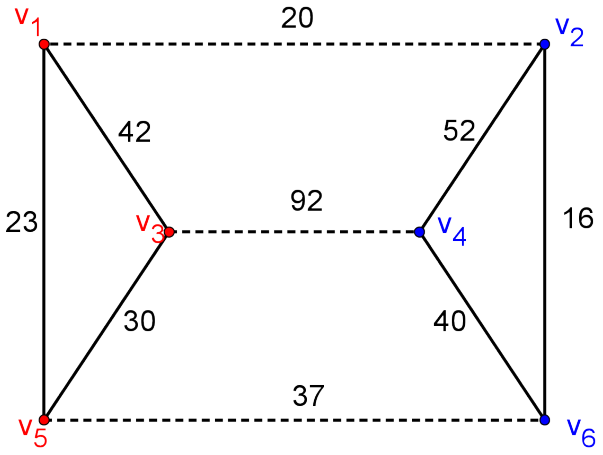
Rozcięcie generowane przez zbiór wierzchołków

Rozcięciem generowanym przez zbiór wierzchołków $W \subseteq V$ nazywamy zbiór krawędzi o jednym końcu w zbiorze W, a drugim w zbiorze $V \setminus W$. Rozcięcie takie oznaczamy symbolem $\delta(W)$:

$$\delta(W) = \{e = (w, v) \colon w \in W, v \in V \setminus W\}$$

Oczywiście $\delta(W) = \delta(V \setminus W)$.

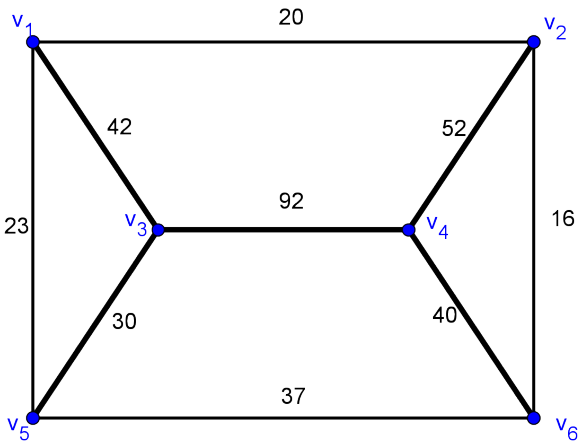


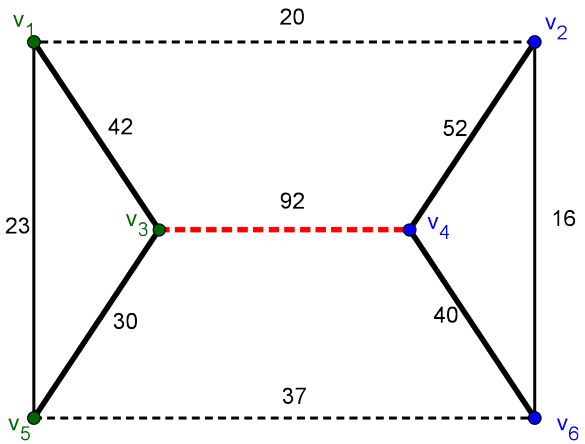


Rozcięcia

Rozcięcie generowane przez krawędź

Niech T będzie drzewem rozpinającym graf G. **Rozcięciem generowanym przez krawędź** $e \in T$ nazywamy rozcięcie (w grafie G) powstałe przez usunięcie krawędzi e (z drzewa T). Rozcięcie takie oznaczamy symbolem $C(e)$.



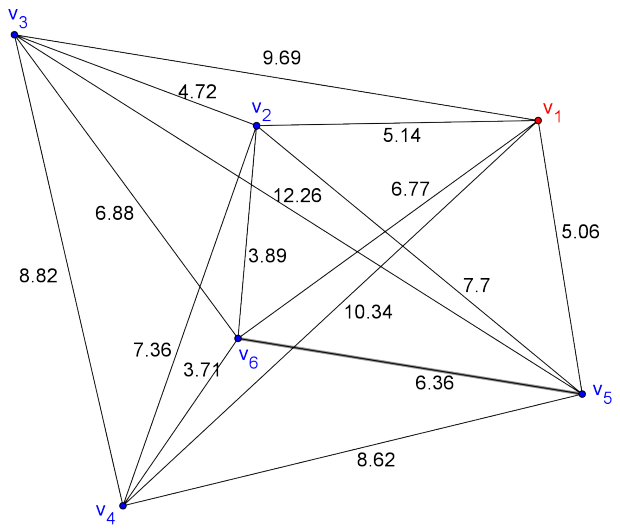


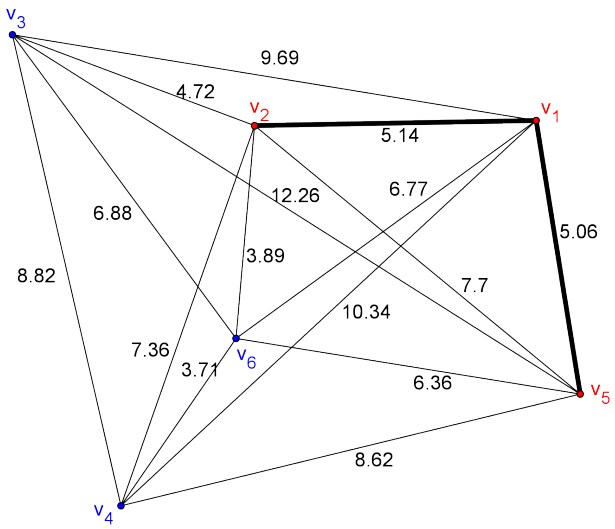
Algorytm Prima

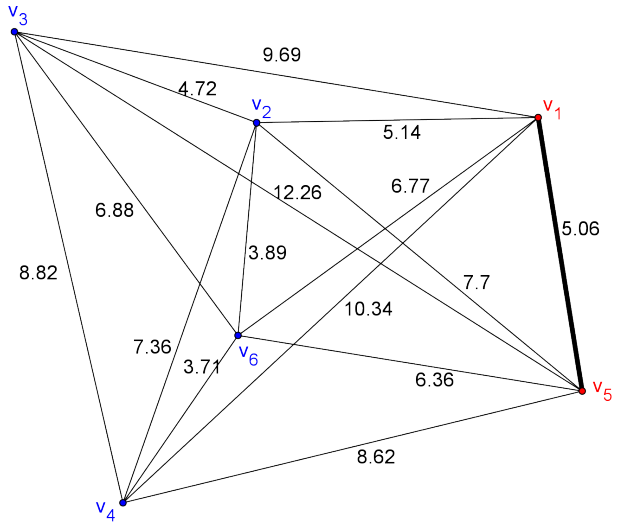
wejście: $G = (V, E, c)$
wyjście: $T \subseteq E$ - minimalne drzewo rozpinające grafu G

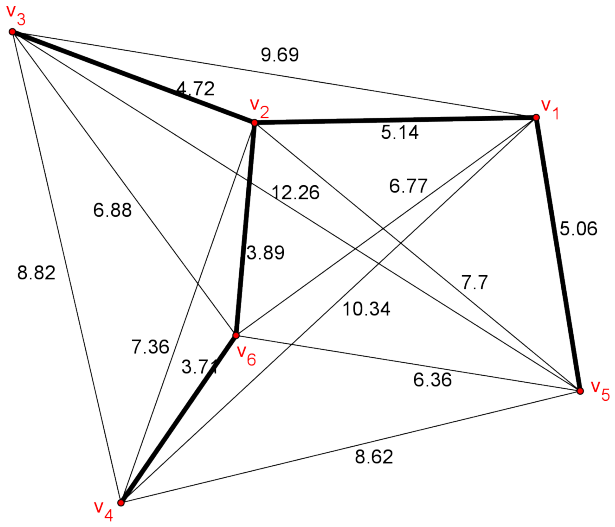
- $W = \{v_0\}$ - dowolny wierzchołek początkowy;
- $T = \emptyset$;
- **for** $(i = 1, \ldots, n-1)$ **do**
  - $e_i = \{v, u\} := \operatorname{argmin}\{c_e \colon e \in \delta(W)\}$; e_i jest krawędzią o najmniejszej wadze "wychodzącą" ze zbioru W; $v \in W$, $u \notin W$
  - $T = T \cup e_i$;
  - $W = W \cup u$;

Przykład działania algorytm Prima









Poprawność algorytmu Prima

Poprawność algorytmu Prima wynika bezpośrednio z poniższego warunku rozcięcia:

Warunek rozcięcia

Rozważmy graf z wagami $G = (V, E, c)$. Drzewo rozpinające $T \subseteq E$ jest minimalne (MST) wtedy i tylko wtedy, gdy

$$\bigwedge_{e \in T} \bigwedge_{f \in C(e)} c_e \leqslant c_f .$$

Dowód warunku rozcięcia

1. Załóżmy, że $T \subseteq E$ jest MST i przypuśćmy, że $\bigvee_{e \in T} \bigvee_{f \in C(e)} c_e > c_f$.
   - $T \cup \{f\}$ zawiera cykl. Usuwając z tego cyklu krawędź e otrzymujemy ST. Nazwijmy je T^*.
   - T^* ma mniejszą wagę niż T, co przeczy temu, że T jest MST.
2. Załóżmy, że ST $T \subseteq E$ spełnia warunek $\bigwedge_{e \in T} \bigwedge_{f \in C(e)} c_e \leqslant c_f$.
   - Niech T^* oznacza takie MST, że $|T^* \cap T|$ jest maksymalna.
   - Przypuśćmy, że $T \neq T^*$ i ustalmy krawędź $e \in T \setminus T^*$.
   - Z warunku rozcięcia $\bigwedge_{f \in C(e)} c_e \leqslant c_f$.
   - T^* jest ST, więc $C(e) \cap T^* \neq \emptyset$. Niech $f' \in C(e) \cap T^*$.
   - $T^* \cup \{e\}$ zawiera cykl. Usuwając z tego cyklu krawędź f' otrzymujemy ST. Nazwijmy je T^{**}.
   - T^{**} jest MST (bo $c_e \leqslant c_{f'}$) i $|T^{**} \cap T| > |T^* \cap T|$, co przeczy maksymalności $|T^* \cap T|$.
   - Zatem $T = T^*$, więc T jest MST.

Warunek ścieżki

W danym drzewie T, niech $P_T(v, w)$ oznacza jedyną ścieżkę łączącą wierzchołki v i w.

Warunek ścieżki

Rozważmy graf z wagami $G = (V, E, c)$. Drzewo rozpinające $T \subseteq E$ jest minimalne (MST) wtedy i tylko wtedy, gdy

$$\bigwedge_{f=\{v,w\}\in E\setminus T} \bigwedge_{e\in P_T(v,w)} c_e \leqslant c_f.$$

Dowód warunku ścieżki

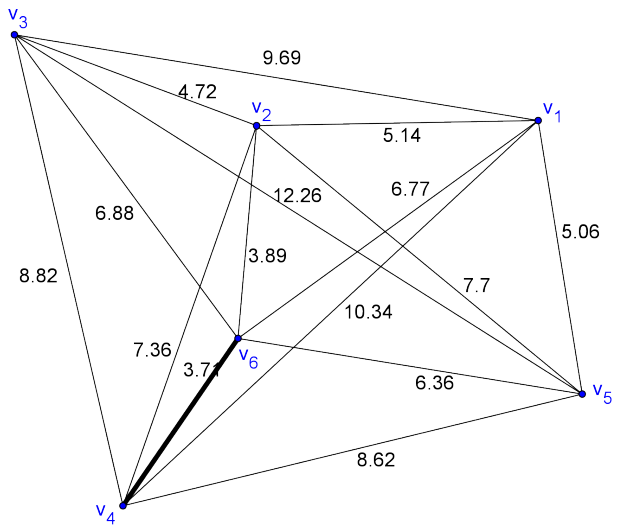
1. Załóżmy, że $T \subseteq E$ jest MST.
   - T spełnia warunek rozcięcia. Ustalmy $f = \{v, w\} \in E \setminus T$ oraz $e \in P_T(v, w)$.
   - Wówczas $f \in C(e)$, a zatem (z warunku rozcięcia) $c_e \leqslant c_f$.
   - T spełnia więc wrunek ścieżki.
2. Załóżmy, że ST $T \subseteq E$ spełnia warunek $\bigwedge_{f=\{v,w\}\in E\setminus T} \bigwedge_{e\in P_T(v,w)} c_e \leqslant c_f$.
   - Ustalmy $e = \{v, w\} \in T$ oraz $f = \{x, y\} \in C(e) \setminus \{e\}$. Oczywiście $f \in E \setminus T$.
   - Zauważmy, że w $T \setminus \{e\}$ nie istnieje ścieżka łącząca x i y. Zatem $e \in P_T(x, y)$.
   - Z warunku ścieżki, $c_e \leqslant c_f$.
   - T spełnia warunek rozcięcia, więc T jest MST.

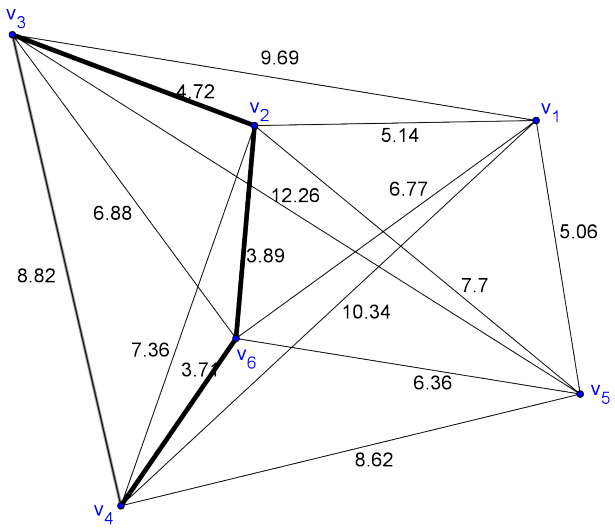
Algorytm Kruskala

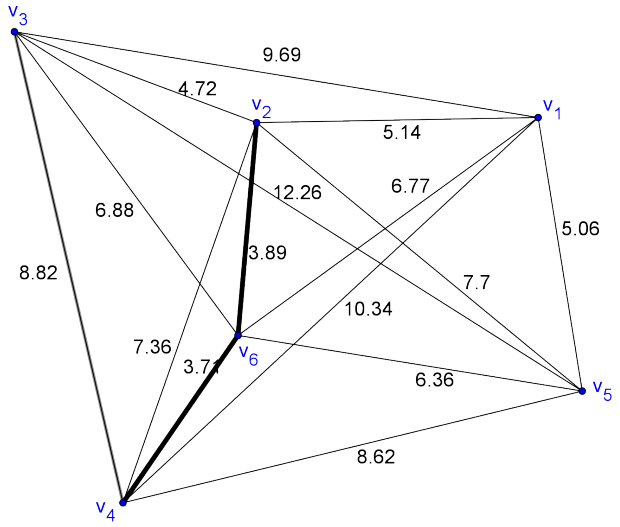
wejście: $G = (V, E, c)$
wyjście: $T \subseteq E$ - minimalne drzewo rozpinające grafu G

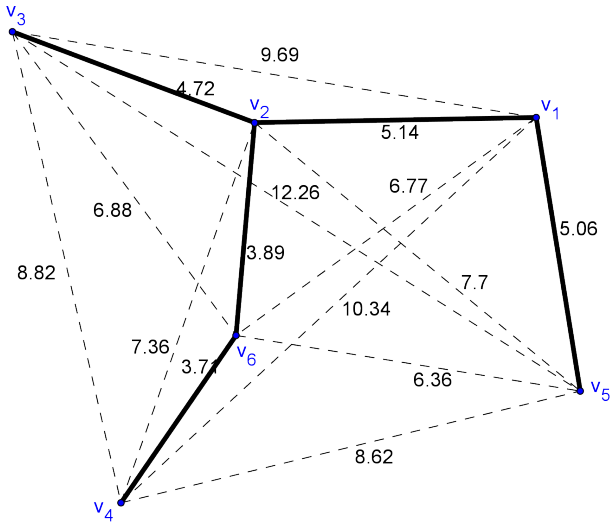
- posortuj krawędzie tak, by $c_{e_1} \leqslant \ldots \leqslant c_{e_m}$;
- $T = \emptyset$;
- **for** $(i = 1, \ldots, m)$ **do**
  - **if** $(T \cup e_i)$ nie zawiera cykli **then**
    - $T = T \cup e_i$;

Przykład działania algorytm Kruskala









Poprawność algorytmu Kruskala

- oczywiście T nie zawiera cykli;
- ze względu na sortowanie, dla każdego zbioru wierzchołków $W \subseteq V$, T zawiera krawędź z rozcięcia $\delta(W)$ o najniższej wadze, więc $T \cap \delta(W) \neq \emptyset$, zatem T jest spójny;
- T jest drzewem
- dowolna krawędź $f = \{v, w\}$ nie należąca do T tworzy cykl z pewnymi krawędziami T - oznaczmy ten cykl $\{v, w\} \cup P_T(v, w)$;
- ze względu na sortowanie, dla każdej krawędzi $e \in P_T(v, w)$ mamy $c_e \leqslant c_f$;
- z warunku ścieżki wnosimy, że T jest MST.

Bibliografia

R. J. Wilson.
Wprowadzenie do teorii grafów.
Wydawnictwo Naukowe PWN, 2008.

M. Uetz.
Introduction / Algorithms & Spanning Trees
Lectures on Discrete Optimization 2010,
http://wwwhome.math.utwente.nl/~uetzm/do/DO_Lecture1.pdf

Dziękujemy za uwagę!