Z4

Na potrzeby czytelniejszego zapisu
grafy traktowane są jako zbiory (czyli z pominięciem odpowiednich
oznaczeń charakterystycznych dla grafów)
Przykładowo jeżeli krawędź należy do grafu to znaczy, że tak naprawdę należy do zbioru krawędzi tego
grafu itp.

$G := (V, E)$

$\forall_{v \in V} \; w(v) \geq 0 \quad (w: E \to \mathbb{R})$

G_i – drzewo powstałe w i-tym kroku algorytmu

T_i – MST w i-tym kroku

Teza: $G_i \subseteq T_i$ ← Pokazujemy to bo w szczególności po n-tym kroku (ostatnim) algorytmu Gn = Tn (ponieważ będzie Gn będzie zawierać się w Tn a jednocześnie będzie miało taką samą liczbę wierzchołków)

Indukcja:

Baza $(n=1)$:

G_1 to drzewo o jednym wierzchołku. Jest jedno MST takiego drzewa, więc $G_1 \subseteq T_1$ ✓

Krok $(G_i \subseteq T_i \Rightarrow G_{i+1} \subseteq T_{i+1})$:

Wykonując kolejny krok algorytmu dodajemy do grafu G_i krawędź $e = (v, v')$ przy czym $v \in G_i \wedge v' \notin G_i$

Przypadki

I) $e \in T_i$, wtedy $G_{i+1} \subseteq T_i = T_{i+1}$

II) $e \notin T_i$.

Skoro T_i to drzewo to istnieje ścieżka p z v do v'.

Ścieżka p zawiera krawędź wychodzącą poza G_i.

Nazwijmy ją $\tilde{e} = (w, w')$, gdzie $w \in G_i$, a $w' \notin G_i$.

Wiemy, że algorytm dobiera krawędzie o minimalnej wadze zatem $w(e) \leq w(\tilde{e})$, wiedząc, że T_i to MST wnioskujemy, że $w(e) = w(\tilde{e})$

$T_{i+1} = (T_i \cup \{e\}) \setminus \{\tilde{e}\}$. Wiemy, że w $T_i \cup \{e\}$ istnieją dwie ścieżki z v do v': p oraz e, czyli w grafie $T_i \cup \{e\}$ $\tilde{e}$ nie jest mostem, a więc jego usunięcie nie rozspójnia grafu.

Co więcej liczba krawędzi między T_i oraz T_{i+1} nie uległa zmianie – T_{i+1} jest nadal drzewem

Ponadto $w(T_{i+1}) = w(T_i)$ $\quad (w(e) = w(\tilde{e}))$
czyli T_{i+1} to MST.

$G_{i+1} \subseteq T_{i+1}$ ✓

Zatem po n-tym kroku $G_n \subseteq T_n$ ale skoro algorytm przebył wszystkie wierzchołki to $G_n = T_n$