Лекція 12. Остовні дерева мінімальної ваги

§3. Остовні дерева

Остовне дерево (підграф) зв’язного графу – це зв’язний ациклічний підграф (дерево), котрий містить всі вершини графа.

Методами пошуку остовного дерева називають алгоритми обходу вершин графа, при якому кожна вершина отримує унікальний порядковий номер.

§4. Остовні дерева мінімальної ваги

Постановка задачі. Задано зв'язний неорієнтований граф $G = (V,E)$, де V –множина вершин, а E – множина ребер і для кожного ребра $(u,v) \in E$ задано вагу $w(u,v)$. Потрібно знайти ациклічну підмножину $T \in E$, яка з'єднує всі вершини і загальна вага якої

$$w(T) = \sum_{(u,v) \in T} w(u,v)$$ мінімальна.

Задача пошуку дерева T називається задачею пошуку мінімального остовного дерева.

Ми розглянемо два алгоритми розв'язку даної задачі – алгоритм Прима та алгоритм Крускала.

3.1 Алгоритм Прима

Ізольованою називається вершина, яка на деякому етапі побудови не зв’язана з іншими вершинами. *Фрагмент* – це підмножина вершин зв’язаних ребрами.

Ізольованим фрагментом називається фрагмент, який на даному етапі побудови не зв’язаний з іншими вершинами або фрагментами.

Принципи побудови дерева мінімальної довжини:

- Довільна ізольована вершина з’єднується з найближчим сусідом – вершиною, яка знаходиться на найменшій відстані від даної вершини.
- Довільний ізольований фрагмент з’єднується з найближчим сусідом найкоротшим ребром.

Алгоритм Прима.

1. Побудова матриці суміжності ваг.

Якщо вершини *u* та *v* не з’єднані, то в матриці на перетині рядка *u* та стовпчика *v* ставиться нескінченість (∞). Діагональні елементи умовно приймаються рівними нескінчености (∞). Всі інші елементи матриці дорівнюють *w*(*u*,*v*).

2. Визначення першого фрагменту.

За початкову обирається довільна вершина. Згідно принципу 1 для цієї вершини знаходимо найближчого сусіда. Для цього в матриці обирається рядок відстаней від обраної вершини до всіх інших і визначається вершина до якої відстань найменша.

3. Розширення фрагменту.

Для розширення фрагменту порівнюються відстані від отриманого фрагменту до кожної ізольованої вершини. З усіх можливих з’єднань обирається така ізольована вершина відстань до якої найменша.

4. Закінчення.

Якщо всі ізольовані вершини приєднані, то мінімальне остовне дерево побудоване (роботу алгоритму завершено), якщо – ні, то перейти на крок 3.

Псевдокод алгоритму Прима:

// Вхідні дані: Зважений зв'язний граф $G=(V,E)$

//Вихідні дані: E_T, множина ребер, які утворюють мінімальне остовне дерево T

$V_T \leftarrow \{ v_0 \}$ //Множина вершин остовного дерева T

$E_T \leftarrow \varnothing$ //Множина ребер остовного дерева T

for $i \leftarrow 1$ **to** $|V| - 1$ **do**

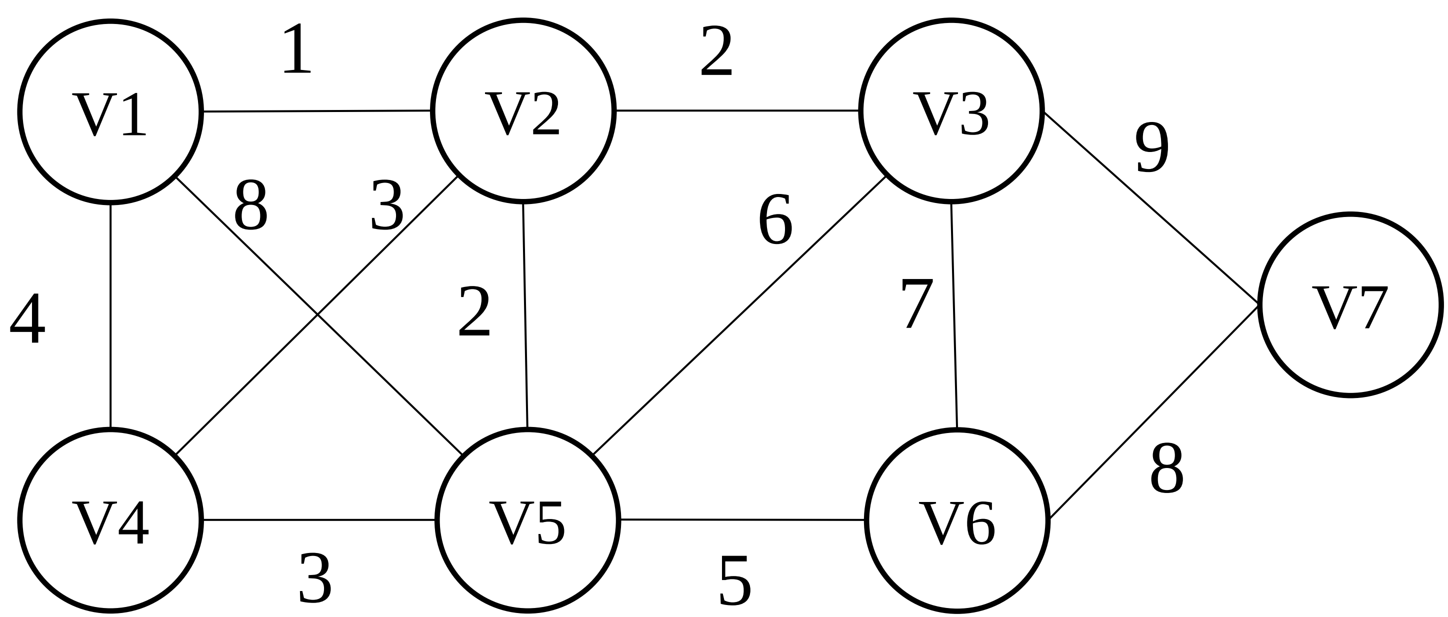
// Пошук ребра з мінімальною вагою $e^* = (u^*, v^*)$ серед всіх ребер (u,v) таких, що $v \in V_T$ та $u \in V - V_T$

$V_T \leftarrow V_T \cup \{u^*\}$

$E_T \leftarrow E_T \cup \{e^*\}$

return E_T

Приклад: Для заданого графа *G* побудувати остовне дерево мінімальної ваги, використовуючи алгоритм Прима.

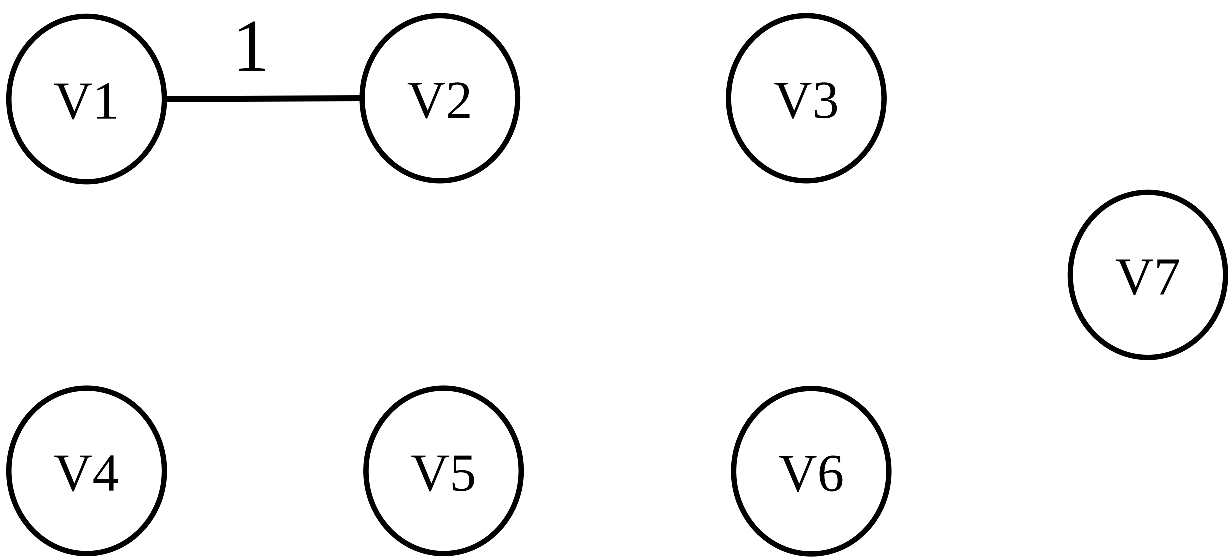


Побудуємо матрицю суміжності ваг W:

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	∞	1	∞	4	8	∞	∞
v_2	1	∞	2	3	2	∞	∞
v_3	∞	2	∞	∞	6	7	9
v_4	4	3	∞	∞	3	∞	∞
v_5	8	2	6	3	∞	5	∞
v_6	∞	∞	7	∞	5	∞	8
v_7	∞	∞	9	∞	∞	8	∞

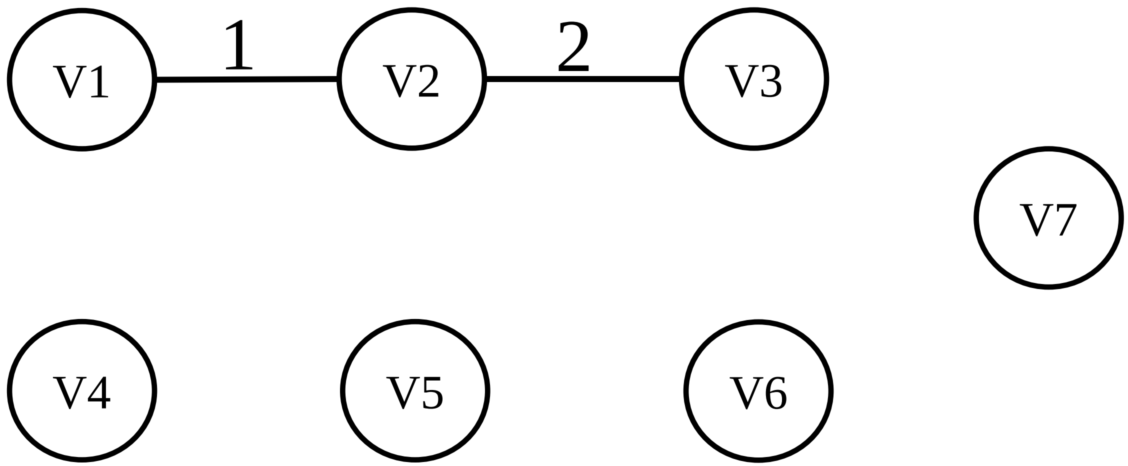
За початкову обираємо довільну вершину, нехай це буде вершина V_1, і для неї шукаємо найближчого сусіда, тобто вершину відстань до якої найменша:

	v_2	v_3	v_4	v_5	v_6	v_7
v_1	1	∞	4	8	∞	∞



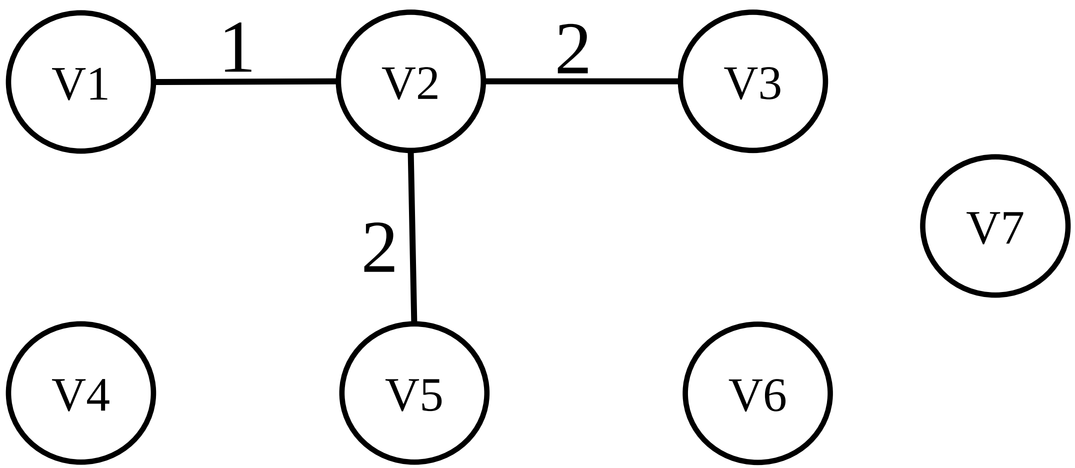
Розширюємо фрагмент:

	v_3	v_4	v_5	v_6	v_7
v_1	∞	4	8	∞	∞
v_2	2	3	2	∞	∞



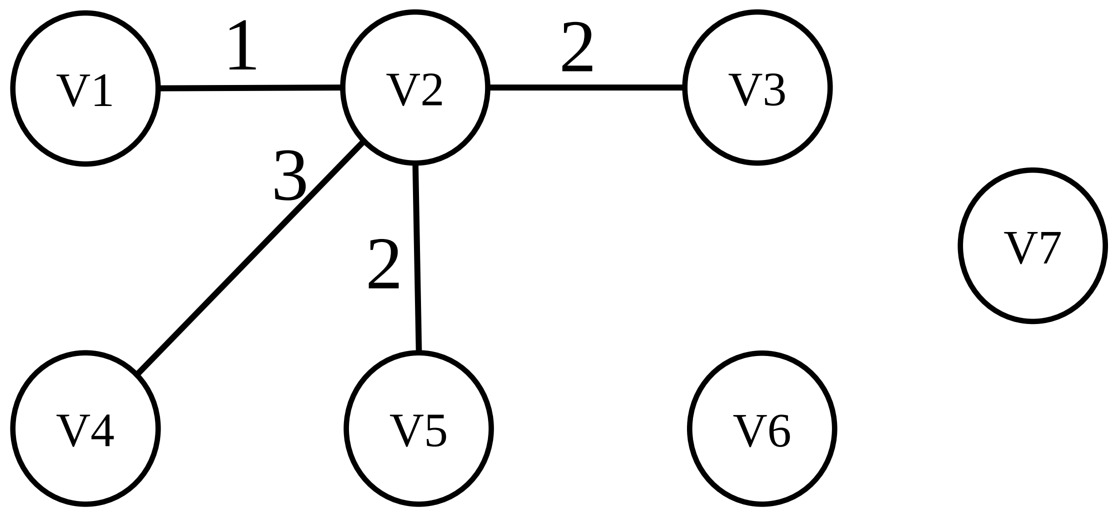
Розширюємо фрагмент:

	v_4	v_5	v_6	v_7
v_1	4	8	∞	∞
v_2	3	2	∞	∞
v_3	∞	6	7	9

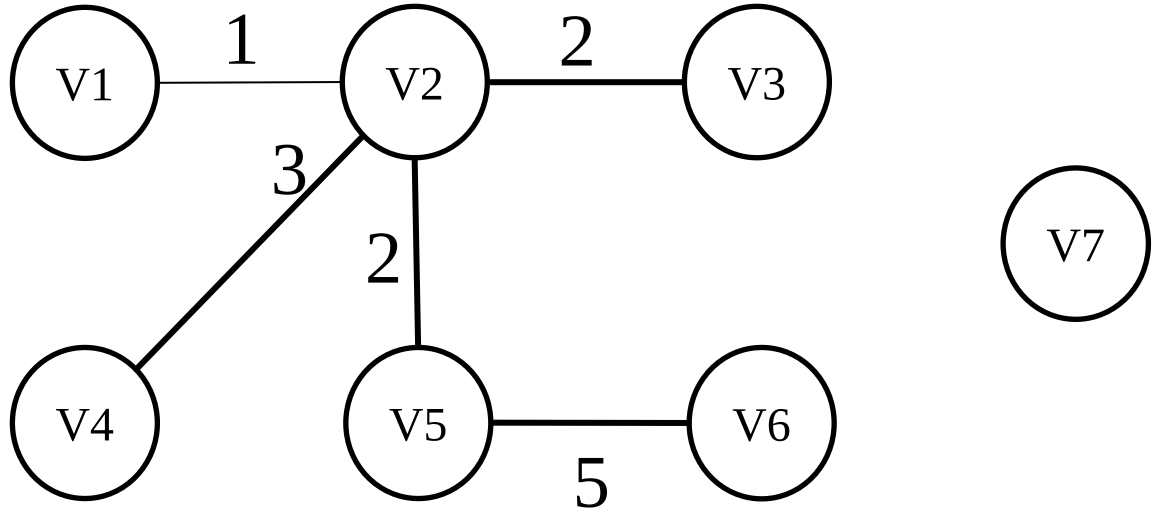


Розширюємо фрагмент:

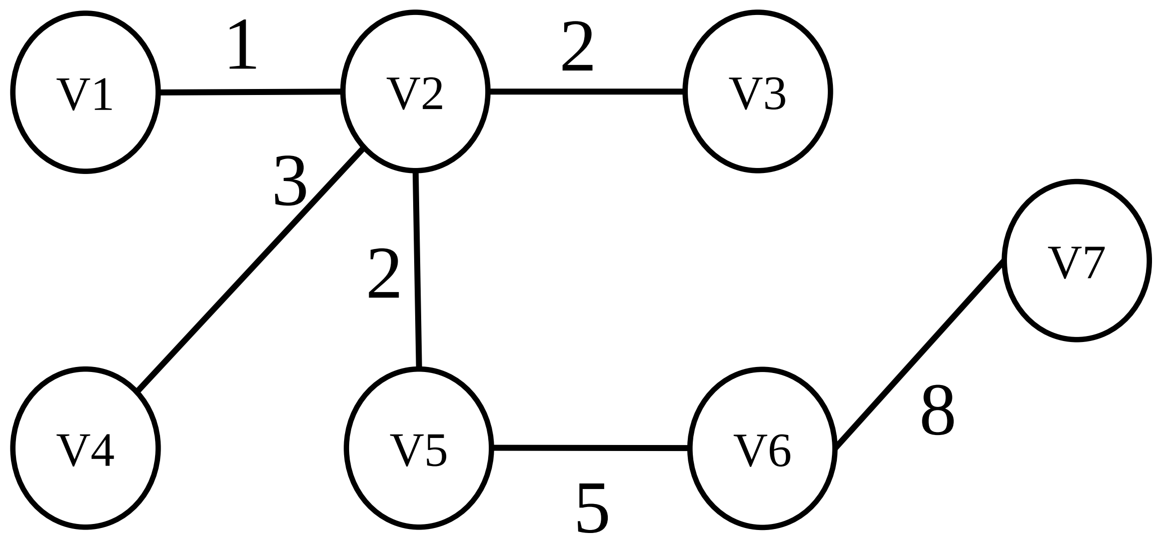
	v_4	v_6	v_7
v_1	4	∞	∞
v_2	3	∞	∞
v_3	∞	7	9
v_5	3	5	∞



	v_6	v_7
v_1	∞	∞
v_2	∞	∞
v_3	7	**9**
v_4	∞	∞
v_5	**5**	∞



	v_7
v_1	∞
v_2	∞
v_3	**9**
v_4	∞
v_5	∞
v_6	**8**



$$w(T) = \sum_{(u,v)\in T} w(u,v) = 1+2+2+3+5+8 = 21$$

3.2 Алгоритм Краскала

Алгоритм Краскала будує мінімальне остовне дерево як послідовність підграфів, котрі завжди ациклічні, але на проміжних стадіях не завжди зв'язні.

Алгоритм Краскала.

1. Відсортувати ребра графу G в зростаючому порядку.

2. Вибрати ребро e_1, яке має в графі G найменшу вагу.

3. На кожному кроці обирати ребро (відмінне від попередніх) з найменшою вагою і таке, що не утворює простих циклів з попередніми ребрами. Отримане дерево T з множиною ребер

$E_T = \{e_1, e_2, e_3, \ldots, e_{n-1}\}$ є мінімальним остовним підграфом графу G.

Псевдокод алгоритму Крускала:

// Вхідні дані: Зважений зв'язний граф $G = (V, E)$

//Вихідні дані: E_T, множина ребер, які утворюють мінімальне остовне дерево T

Сортування множини E за зростанням ваг ребер $w(e_1) \leq \ldots \leq w(e_n)$, $n = |E|$

$E_T \leftarrow \varnothing$ //Множина ребер остовного дерева T

$ecounter \leftarrow 0$//розмір дерева T

$k \leftarrow 0$ //кількість оброблених ребер

while $ecounter < |V| - 1$ **do**

$k \leftarrow k + 1$

  if $E_T \cup \{e_k\}$ – ациклічний граф **then**

  $E_T \leftarrow E_T \cup \{ e_k \}$; $ecounter \leftarrow ecounter + 1$

return E_T

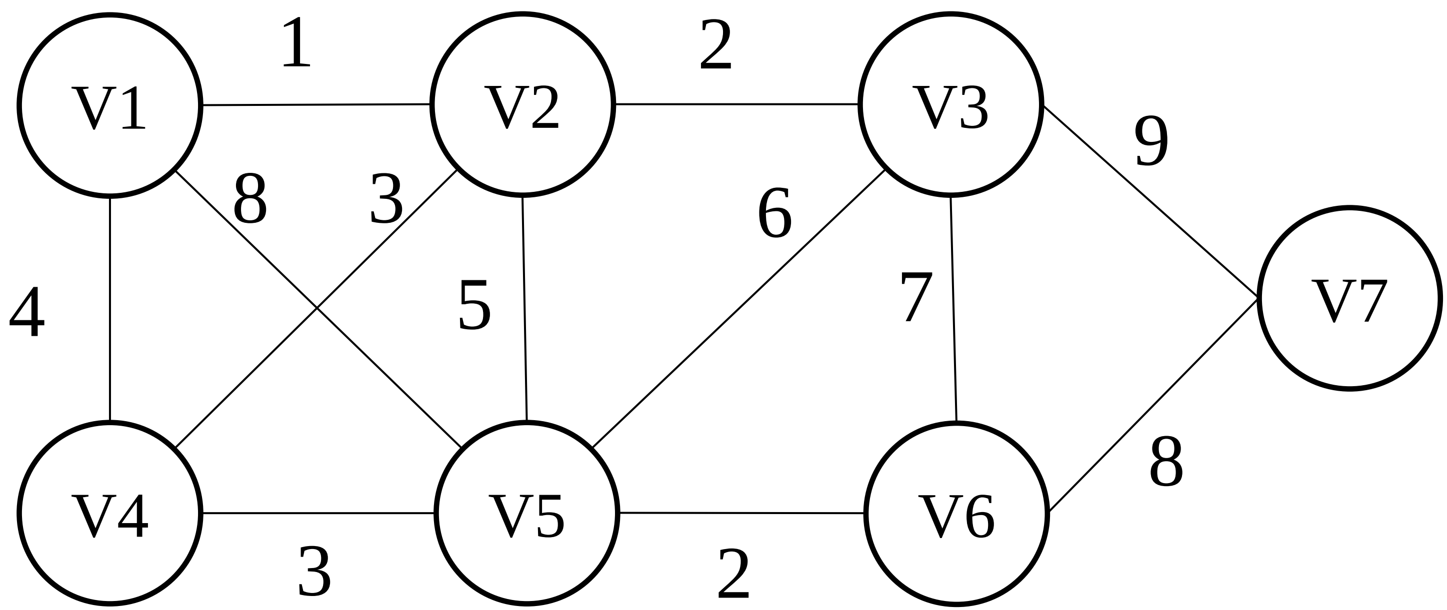
Одна з можливих реалізацій алгоритму Крускала.

Початок. Упорядкувати множину ребер за зростанням ваг: e_1, e_2, e_3, ..., e_n. Виконати розбиття множини вершин $\rho = \{\{v_1\}, \{v_2\}, ..., \{v_m\}\}$

Ітерація. Вибрати таке чергове ребро з упорядкованої послідовності ребер, кінці якого містяться в різних множинах розбиття. Нехай обрано ребро $e_i(u,v)$, тоді множини, що містять вершини u та v об'єднуються в одну множину.

Закінчення. Роботу закінчити, коли буде вибране $(n-1)$ ребро, при цьому всі підмножини розбиття об'єднуються в одну.

Приклад. Для заданого графа *G* побудувати остовне дерево мінімальної ваги, використовуючи алгоритм Крускала.



Ребра впорядковані за зростанням	Розбиття множини вершин	Вибір ребра у мінімальний остов T
	$\rho_0=\{\{v_1\},\{v_2\},\{v_3\},\{v_4\},\{v_5\},\{v_6\},\{v_7\}\}$	
$e_1(v_1, v_2) = 1$	$\rho_1=\{\{v_1,v_2\},\{v_3\},\{v_4\},\{v_5\},\{v_6\}, \{v_7\}\}$	$E_T = \{ e_1 \}$
$e_2(v_2, v_3) = 2$	$\rho_2=\{\{v_1,v_2,v_3\},\{v_4\},\{v_5\},\{v_6\}, \{v_7\}\}$	$E_T = \{ e_1, e_2\}$
$e_3(v_5, v_6) = 2$	$\rho_3=\{\{v_1,v_2,v_3\},\{v_4\},\{v_5,v_6\}, \{v_7\}\}$	$E_T = \{ e_1, e_2, e_3\}$
$e_4(v_2, v_4) = 3$	$\rho_4=\{\{v_1,v_2,v_3,v_4\},\{v_5,v_6\}, \{v_7\}\}$	$E_T = \{ e_1, e_2, e_3, e_4\}$
$e_5(v_4, v_5) = 3$	$\rho_5=\{\{v_1,v_2,v_3,v_4,v_5,v_6\}, \{v_7\}\}$	$E_T = \{ e_1, e_2, e_3, e_4, e_5\}$
$e_6(v_1, v_4) = 4$	ρ_5	$e_6 \notin E_T$
$e_7(v_2, v_5) = 5$	ρ_5	$e_7 \notin E_T$
$e_8(v_3, v_5) = 6$	ρ_5	$e_8 \notin E_T$
$e_9(v_3, v_6) = 7$	ρ_5	$e_9 \notin E_T$
$e_{10}(v_1, v_5) = 8$	ρ_5	$e_{10} \notin E_T$
$e_{11}(v_6, v_7) = 8$	$\rho_6=\{\{v_1,v_2,v_3,v_4,v_5,v_6,v_7\}\}$	$E_T = \{ e_1, e_2, e_3, e_4, e_5, e_{11}\}$
$e_{12}(v_3, v_7) = 9$	-	-

При приєднанні ребра e_{11} робота алгоритму закінчується, так як вже приєднано

$n-1= 7-1=6$ ребер і всі підмножини розбиття об'єдналися в одну $\rho_6=\{\{v_1,v_2,v_3,v_4,v_5,v_6,v_7\}\}$. Остовне дерево мінімальної ваги T утворюють ребра e_1, e_2, e_3, e_4, e_5, e_{11} .