ДЕРЕВА
МІНІМАЛЬНІ ОСТОВНІ ДЕРЕВА ЗВАЖЕНИХ ГРАФІВ

1. Дерева

Означення. ***Деревом*** *називається зв'язний неорієнтований граф без циклів.* Дерево не містить петель і кратних ребер.

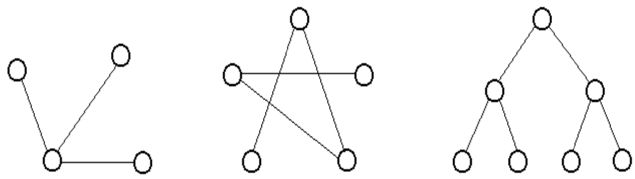
Цьому означенню еквівалентні, як легко показати, наступні твердження:

а) дерево є зв'язний граф, що містить n вершин і $n-1$ ребер;

б) дерево є граф, будь-які дві вершини якого можна з'єднати простим ланцюгом.

в) дерево є граф без циклів, додаючи до якого нове ребро можна дістати один простий цикл.

Приклад 7. Графи, зображені на малюнках є деревами.



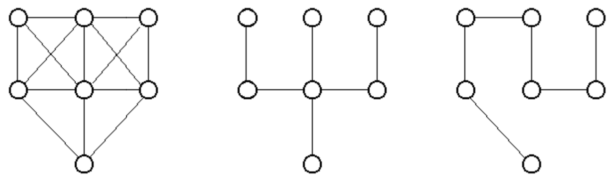
а) б) в)

Означення. ***Лісом*** *називається незв'язний неорієнтований граф без циклів, в якому кожна компонента зв'язності є деревом.*

Приклад 8. Малюнок з прикладу 7 можна розглядати як ліс з трьох дерев.

Означення. ***Остовним деревом*** (spanning tree) *для графа* $G=(V,E)$ *називається остовний підграф (тобто підграф, який містить всі вершини графа* G*), який є деревом.*

Приклад 8. Для графа на малюнку а) графи б) і в) є остовними деревами:



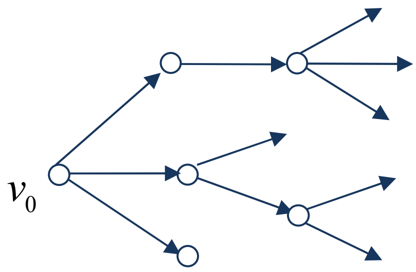
а) б) в)

Будь-яка частина дерева або ліса також є деревом або лісом. Будь-який ланцюг у такому графі простий (інакше він містив би цикл).

Нехай граф G має n вершин і m ребер Оскільки всяке дерево з n вершинами за означенням має $n-1$ ребер, то будь-яке остовне дерево графа G виходить з цього графа в результаті видалення $m-(n-1)=m-n+1$ ребер.

Означення. *Число* $g=m-n+1$ *називається* ***цикломатічним числом*** *графа.*

Якщо в дереві G виділено якусь вершину v_0, то цю вершину називають ***коренем*** дерева G, а саме дерево називають ***деревом з коренем***. У дереві з коренем можна природним чином орієнтувати ребра. Вершину v' ребра (v', v'') можна з'єднати єдиним ланцюгом з коренем v_0. Якщо цей ланцюг не містить ребра (v', v''), то вводиться орієнтація від v' к v'', в противному випадку – від v'' до v'. Орієнтоване в такий спосіб дерево з коренем називається орієнтованим деревом. У ньому всі ребра мають напрямок від кореня:



У кожну вершину орієнтованого дерева (за винятком v_0) входить тільки одне ребро, тобто, ця вершина є кінцем одного і тільки одного ребра. У корінь не входить жодне ребро, усі інцидентні кореню ребра зв'язують його зі своїми другими кінцями, виходить, v_0 є їхнім початком.

Будь-яке дерево можна орієнтувати, вибравши як корінь будь-яку його вершину.

2. Мінімальні остовні дерева зважених графів

<u>Означення.</u> *Граф* $G = (V, E)$ *називається* ***зваженим****, якщо кожному ребру* (v_i, v_j) *зіставлене деяке число* $c(v_i, v_j)$*, яке називається його* ***довжиною*** *(або* ***вагою****, або* ***вартістю****).*

<u>Означення.</u> ***Матрицею довжин ребер*** *або* ***матрицею вагів*** *графа* $G = (V, E)$ *називається матриця* $C(G) = \left\{c_{ij}, i, j = \overline{1, n}\right\}$*, де*

$$c_{ij} = \begin{cases} c(v_i, v_j), \text{ якщо існує ребро з вершини } v_i \text{ у вершину } v_j; \\ \infty, \text{ в протилежному випадку.} \end{cases}$$

Матриця довжин ребер неорієнтованого графа є симетричною.

Нехай G – зв'язний зважений граф. Задача побудови *мінімального остовного дерева* (*minimal spanning tree*) полягає в тому, щоб в множині остовних дерев знайти дерево, в якого сума довжин ребер мінімальна.

Необхідність побудови мінімального остовного дерева графа виникає, наприклад, у типових випадках, коли

а) Потрібно з'єднати n міст комунікаційними лініями (залізничними лініями, автомобільними дорогами, лініями електропередач, мережею трубопроводів і т. д.) так, щоб сумарна довжина ліній або їх вартість була б мінімальною.

б) Потрібно побудувати схему електричної мережі, в якої клеми повинні бути сполучені за допомогою проводів найменшої загальної довжини.

Для побудови мінімального остовного дерева, яке має своїм коренем одну з вершин будь-якого зваженого графа, можуть бути використані методи Краскала (Joseph Bernard Kruskal (1928 – 2010) – американський математик), Пріма (Robert Clay Prim (1921) – американський математик) або Борувки (Otakar Borůvka (1899–1995) – чеський математик). Ці алгоритми відповідають т.з. «жадібній» стратегії: на кожному кроці вибирається локально найкращий варіант.

Розглянемо більш детальніше алгоритм Краскала (1956). Алгоритм Краскала спочатку поміщає кожну вершину в своє дерево, а потім поступово об'єднує ці дерева, об'єднуючи на кожному кроці два деяких дерева деяким ребром. Перед початком виконання алгоритму, усі ребра сортуються за довжиною (в порядку неспадності). Потім починається процес об'єднання: перебираються всі ребра від першого до останнього (у порядку сортування), і якщо в поточного ребра його кінці належать різним піддеревам, то ці піддерева об'єднуються, а ребро додається до відповіді. Після закінчення перебору всіх ребер всі вершини будуть належати одному піддереву, і відповідь знайдено. Підграф даного графа, який містить всі його вершини і знайдену множину ребер, є його минимальним остовним деревом.

Алгоритм Краскала

***Крок* 0.** Установка початкових значень.

Вводимо матрицю довжин ребер $C(G)$ графа G.

Крок 1. Вибираємо в графі G ребро мінімальної довжини (якщо таких ребер декілька, беремо будь-яке з них). Будуємо граф G_1, що складається з даного ребра і інцидентних йому вершин. Оскільки $i \neq n$, то переходимо до кроку 2.

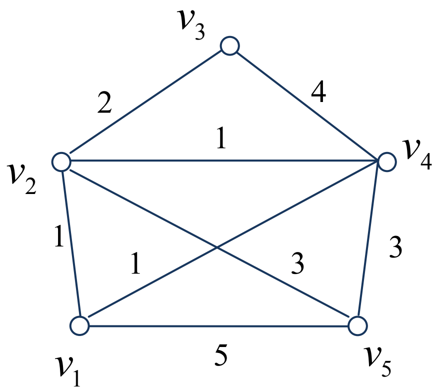
Крок i для будь-якого $i > 1$. Побудувати граф G_i, додаючи до графа G_{i-1} нове ребро мінімальної довжини, вибране серед всіх ребер графа G, кожне з яких інцидентне якій-небудь вершині графа G_{i-1} і одночасно інцидентне якій-небудь вершині графа G, що не міститься в G_{i-1}. Разом з цим ребром включаємо в G_i й інцидентну йому вершину, що не міститься в G_{i-1}. Якщо $i = n$, де $n = |E|$ – число ребер графа, то граф G_i – шукане мінімальне остовне дерево (задача розв'язана), якщо $i \neq n$ – перейти до кроку ***i***+1.

<u>Зауваження.</u> Виконання алгоритму Краскала можна завершити відразу ж, як тільки в дерево буде додано $(n-1)$-е ребро (оскільки в дереві з n вершинами має бути точно $n-1$ ребро).

Можна довести, що *якщо в початковому графі число вершин дорівнює* n, *то підграф* G_{n-1} *буде шуканим остовним деревом.*

Розглянемо роботу алгоритму Краскала на прикладі.

<u>Приклад 9</u>. Знайти мінімальне остовне дерево для графа, зображеного на малюнку.

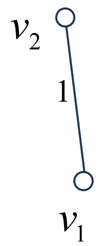


<u>Розв'язання</u>.

***Крок* 0.** Вводимо матрицю довжин ребер $C(G)$ графа G.

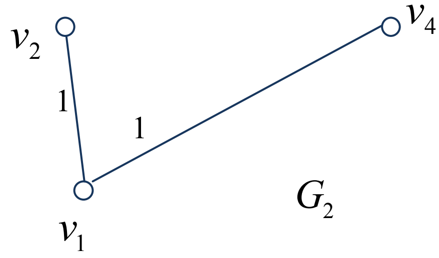
$$C(G)=\begin{pmatrix} \propto & 1 & \propto & 1 & 5 \\ 1 & \propto & 2 & 1 & 3 \\ \propto & 2 & \propto & 4 & \propto \\ 1 & 1 & 4 & \propto & 3 \\ 5 & 3 & \propto & 3 & \propto \end{pmatrix}$$

Крок 1. Вибираємо в графі G ребро мінімальної довжини. Ребер мінімальної довжини 1 три: (v_1, v_2), (v_1, v_4), (v_2, v_4). Беремо (v_1, v_2). Будуємо граф G_1, що складається з даного ребра і інцидентних йому вершин. Покладаємо $1 = i \neq n = 5$. Оскільки $i \neq n = 5$, то переходимо до кроку 2.

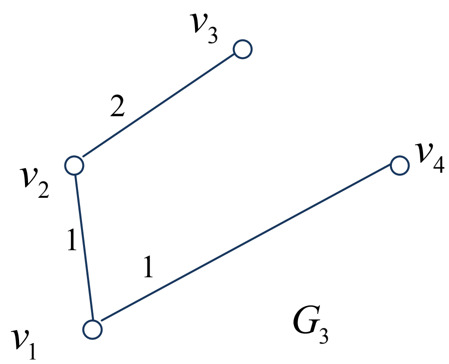


G_1

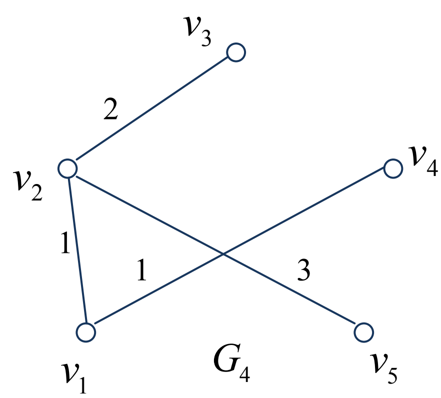
Крок 2. Будуємо граф G_2, додаючи до графа G_1 нове ребро мінімальної довжини, вибране серед всіх ребер графа G, кожне з яких інцидентне одній з вершин v_1, v_2 графа G_1 і одночасно інцидентне якій-небудь вершині графа G, що не міститься в G_1, тобто одній з вершин v_3, v_4, v_5. Таким чином, треба вибрати ребро мінімальної довжини з ребер (v_1, v_4), (v_1, v_5), (v_2, v_3), (v_2, v_4), (v_2, v_5). Ребер мінімальної довжини 1 два: (v_1, v_4), (v_2, v_4). Беремо (v_1, v_4). Разом з цим ребром включаємо в G_2 й інцидентну йому вершину v_4, що не міститься в G_1. Покладаємо $i = 2$. Оскільки $2 = i \neq n = 5$, то переходимо до кроку 3.



***Крок* 3.** Будуємо граф G_3, додаючи до графа G_2 нове ребро мінімальної довжини, вибране серед всіх ребер графа G, кожне з яких інцидентне одній з вершин v_1, v_2, v_4 графа G_2 і одночасно інцидентне якій-небудь вершині графа G, що не міститься в G_2, тобто одній з вершин v_3, v_5. Таким чином, треба вибрати ребро мінімальної довжини з ребер (v_1, v_5), (v_2, v_3), (v_2, v_5), (v_4, v_5). Ребро мінімальної довжини 2 одне: (v_2, v_3). Разом з цим ребром включаємо в G_3 й інцидентну йому вершину v_3, що не міститься в G_2. Покладаємо $i = 3$. Оскільки $3 = i \neq n = 5$, то переходимо до кроку 4.



***Крок* 4.** Будуємо граф G_4, додаючи до графа G_3 нове ребро мінімальної довжини, вибране серед всіх ребер графа G, кожне з яких інцидентне одній з вершин v_1, v_2, v_3, v_4 графа G_3 і одночасно інцидентне вершині графа G, що не міститься в G_3, тобто вершині v_5. Таким чином, треба вибрати ребро мінімальної довжини з ребер (v_1, v_5), (v_2, v_5), (v_4, v_5). Ребер мінімальної довжини 3 два: (v_2, v_5), (v_4, v_5). Беремо (v_2, v_5).



Разом з цим ребром включаємо в G_4 й інцидентну йому вершину v_5, що не міститься в G_3. Покладаємо $i = 4$. Оскільки $4 = i = n - 1$, то граф G_4 – шукане мінімальне остовне дерево.