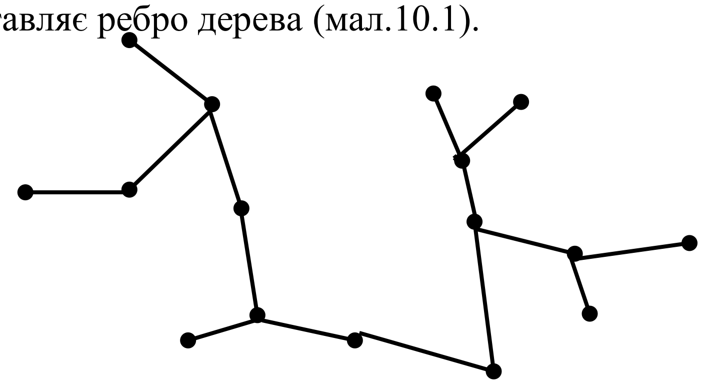
ЛЕКЦІЯ 10. ЕВКЛІДОВЕ МІНІМАЛЬНЕ КІСТЯКОВЕ ДЕРЕВО(ЕМКД)

3.1 ОЗНАЧЕННЯ ТА ВИКОРИСТАННЯ ЕМКД

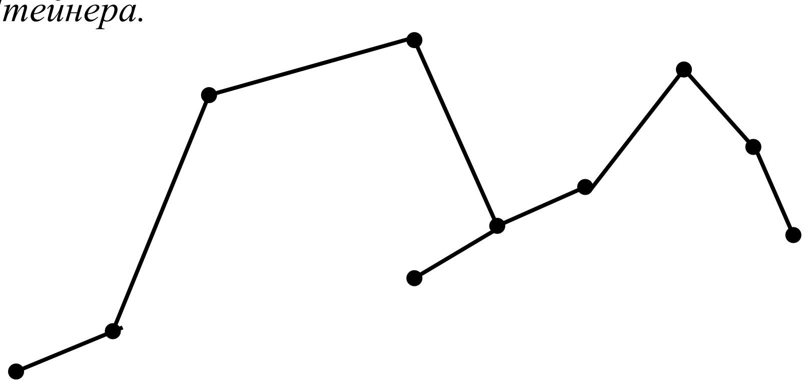
Задача Б.3 (ЕВКЛІДОВЕ МІНІМАЛЬНЕ КІСТЯКОВЕ ДЕРЕВО). *На площині задано N точок. Побудувати дерево, вершинами якого є усі задані точки і сумарна довжина усіх ребер якого мінімальна.*

Розв'язком задачі є список, який містить N-1 пару точок. Кожна пара представляє ребро дерева (мал.10.1).

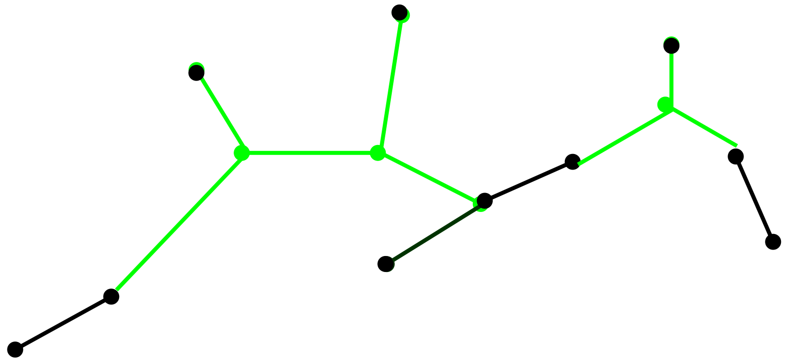


Мал.10.1. Мінімальне остове дерево множини точок на площині.

Означення 10.1. *Дерево, яке має найкоротшу довжину і задовольняє умові, що до вихідної множини не заборонено додавати нові точки, називається деревом Штейнера.*



(а)



(б)

Мал.10.2 а,б. Сумарна довжина ребер дерева Штейнера (б) може бути меншою, ніж у МОД.

ЕМКД застосовують при кластеризації та розпізнаванні образів. Використовувалось воно для мінімізації довжини провідниківпри компоновці схеми ранніх ЕОМ. Мінімальне остове дерево дає початкове наближення для багатьох алгоритмів розв'язання задачі про комівояжера. Перспективною можна важати ідею побудови на основі ЕМКД сучасних комп'ютерних мереж.

Задача про МОД. (задачача теорії графів): *задано зважений граф G(N,E) з N вершинами і E ребрами, необхідно знайти найкоротше піддерево графа G, яке включає усі його вершини.*

Насьогодні найшвидше задача може бути розв'язана за час O(E), де $E > N^{1+\varepsilon}$. Так, як у випадку МКД кількість вхідних ребер, у гіршому випадку, досягатиме N(N-1)/2, а вага ребра дорівнює відстані між точками, з'єднаних ребром, то використання, в цьому випадку, кращого із відомих алгоритмів розв'язання задачі про МКД вимагатиме часу $\theta(N^2)$. Таким чином нижньою оцінкою задачі про МКД буде $\Omega(N^2)$.

У випадку евклідової задачі про МКД N вершин визначаються 2N координатами точок на площині, а відповідний граф містить ребра, які з'єднують кожну пару вершин. Відповідно такий варіант задачі містить сильні обмеження у порівнянні з загальним випадком, а тому є сподівання ,що у евклідовому випадку задача про МКД може бути розв'язана за коротший час.

3.2 РОЗВ'ЯЗОК ЗАДАЧІ ПРО ЕМКД ЗА ДОПОМОГОЮ ДІАГРАМИ ВОРОНОГО

Оскільки задача сортування зводиться до ЕМКД за лінійний час. Звідкии слідує, що нижня оцінка для часу побудови ЕМКД множини із N точок буде Ω(NlogN). Із діаграми зводи мості задач близкості випливає, що задача ЕМКД зводиться до Діаграми Вороного за лінійний час, і тому виникає ідея методу побудови ЕМКД за допомогою ДВ.

Лема 10.1. *Нехай G=(V,E) – є зважений неорієнтований граф. Якщо має місце розбиття цього графа на підмножини вершин* $V=V_1 \cup V_2$*.Тоді в G існує МКД мінімальної вартості таке, що одна вершина* $u \in V_1$ *, а інша* $v \in V_2$*.*

Лема 10.1а: *Нехай G=(V,E) – зв'язний неорієнтований граф та* G^* *=(V,T) – остове дерево мінімальної вартості (тут вартість є відстань). Тоді для будь-яких вузлів з V шлях між ними в S – єдиний. Якщо до S додати ребро з E\T, то виникне рівно один цикл.*

Лема 10.1б: *Нехай G=(V,E) – зв'язний неорієнтований граф та C – функція вартості, задана на його ребрах (відстань). Нехай {(V1,T1), (V2,T2),…, (Vк,Тк)} – довільний остовий ліс для G, к>1, T – обєднання Ti, i=1..к. Припустимо, що t = (v, w) – ребро найменшої вартості в E\T, причому v належитьV1 і w не належить V1. Тоді знайдеться остове дерево для G, що містить T та {e}, вартість якого не більша вартості любого остового дерева для G, яке містить T.*

Множина вершин V – це множина точок S, а довжина ребра дорівнює евклідовій відстані між його кінцевими точками. На кожному кроці побудови ЕМКД алгоритм буде опрацьовувати ліс, який міститеме кілька дерев (вони у результаті стануть піддеревами остаточного ЕМКД). На початку ліс представлений сукупністю вершин, тобто кожна точка є окремим деревом без ребер. Основний крок алгоритму виконується наступним чином (через d(u, v) позначимо довжину відрізка uv):

1. Вибрати із лісу деяке дерево T.
2. Знайти ребро (u′, v′) таке, щоб d(u′, v′) = min{d(u, v): u ∈ T, v ∈ S – T}
3. Якщо T′ – дерево, яке містить вершину v′, то об'єднати T і T′, з'єднавши їх ребром (u′, v′).

Виконання алгоритму припиняється, коли ліс містить єдине дерево, яке і є ЕМКД початкової множини.

Найскладнішим у даному алгоритмі буде крок 2, а тому скористаємось діаграмою Вороного та двоїстою до неї триангуляцією Делоне.

Лема 10.2. *Нехай S – множина точок на площині, а Δ (p) –множина точок суміжних з $p \in S$ у триангуляції Делоне множини S. Тоді, для довільного розбиття множини $S = S_1 \cup S_2$ найкоротший відрізок (p,q), який з'єднує точки $p \in S_1$, $q \in S_2$ належитиь тріангуляції Делоне і $q \in \Delta (p)$.*

Відповідно до цієї леми досить розглянути лише ребра триангуляції Делоне, тобто необхідно шукати МКД планарного графа.

Для вибору дерева T (що об'єднається із іншим деревом лісу) було запропоновано (Черитоном і Тар'яном) однорідне правило вибору, перед використанням якого усі дерева, які містять по одній вершині, поміщуються у чергу.

1.Вибрати дерево із початку черги.

2.Якщо дерево T′′ отримали в результаті об'єднання дерева T із деяким іншим деревом T′, то вилучити T і T′ із черги і добавити T′′ у кінець черги.

Оскільки щоразу відбувається об'єднання двох дерев і початкова черга містить N дерев, то основний крок алгоритму виконається рівно (N-1) разів.

<u>Алгоритм</u>

1.На заданій множині *S* побудувати Діаграму Вороного і відповідну їй триангуляцію Делоне (лема 10.2) (*O(NlogN)*).

2.Сформувати ліс дерев, що складаються із однієї точки (створити чергу) *O(N)*.

1. Поки ліс не перетвориться в єдине дерево виконувати наступні дії *O(N)*:

1) вибрати дерево *T* з початку черги;

2) знайти дерево *T'* таке, що *T'* містить вершину, яка задовольняє лемі 10.2;

3) об'єднати *T* і *T'* в *T'';*

4) вилучити *T, T'* з черги , а *T''*помістити в кінець черги.

Теорема 10.1. *ЕМКД множини S із N точок на площині може бути побудовано за допомогою тріангуляції Делоне(ДВ) за оптимальний час $\theta(N\log N)$.*

Реалізацію алгоритма можна реалізувати поетапно.
Визначається для кожного дерева із черги T ціле число, яке можна назвати, наприклад, *Етапом*, і яке обчислюється наступним чином:

Етап(T) = 0, якщо T=1,

Етап(T) = min(*Етап*(T′), *Етап*(T′′))+1, якщо T є результатом об’єднання дерев T′ і T′′.

Черга дерев має цікаву властивість: у будь-який момент виконання алгоритму сукупність чисел *Етап*(T), у напрямку від початку до кінця черги, утворює неспадну послідовність. *Етап* j вважається завершеним, якщо із черги вилучено дерево T_j, для якого *Етап*(T) = j і ні для якого іншого дерева T′ у черзі значення *Етап*(T′) не дорівнює j. Варто відзначити, що після закінчення *Етапу* j черга містить не більше $N/2^{j+1}$ елементів, а всього інує не більше $[\log_2 N]$ етапів. Подальше опрацювання відбувається наступним чином. Після вибору із черги першого елементу T (при цьому *Етап*(T) = j) серед ребер, які з’єднують вершини дерева T із вершинами, які йому не належать, знаходимо ребро найменшої вартості. При такій організації опрацювання на кожному етапі кожне ребро переглядається не більше двох разів (за винятком ребер, що уже належать T). Отже кожен етап опрацювання можна виконати за час, пропорційний кількості ребер, яка завдяки планарності графу дорівнює O(N). Оскільки число етапів не перевищує $[\log_2 N]$, то для часу виконання алгоритму отримуємо оцінку O(N logN).