# ЛЕКЦІЯ 1

## *ВСТУП*

1. *Означення.* Обчислювальна геометрія - це наука предметом дослідження, якої є аналіз та побудова ефективних алгоритмів розв'язання геометричних задач, оцінки їх складності в рамках теорії алгоритмів.
2. *Застосування.*

Обчислювальна геометрія має широке коло застосувань. Це робототехніка, проектування СВІС , бази даних, дослідження операцій , теорія керування, комп'ютерна графіка.

3. *Основні класи задач:*

випуклість, перетин, геометричний пошук, близькість, оптимізація.

*По способу обробки виділяють три типи задач*:

пошук підмножини, обчислення, розпізнавання.

4. *Приклади задач які розв'язує ОГ.*
   - задача комівояжера на евклідовій матриці, побудова мінімального кістякового дерева, вилучення невидимих ліній, лінійне програмування.
5. *Напрямки обробки зображувальної інформації:*

розпізнавання образів, обробка зображень, комп'ютерна графіка.

1)*Основна задача розпізнавання образів* полягає у перетворені уже існуючого зображення на формальну зрозумілу мову символів. Розпізнавання образів є сукупність методів, які дозволяють отримати опис зображення поданого на вхід, або віднести задане зображення до якогось класу. (сортування пошти, медична діагностика і томографія)

2) *Обробка зображень* розглядає задачі, в яких вхідні і вихідні дані є зображеннями.( передача зображень разом із погашенням шумів і стискання даних, перехід від одного типу зображень(напівтонового) до іншого ( каркасного), контрастування різних знімків, також синтез існуючих зображень у нові).

3)*Комп'ютерна графіка* створює зображення у випадку, коли початковою є інформація не зображувальної природи. Наприклад, візуалізація експериментальних даних у вигляді графіків, гістограм чи діаграм, вивід інформації на екран у комп'ютерних іграх, синтез сцен для тренажерів.

6. *Напрями комп'ютерної графіки:*

1). *Ілюстративний*, який можна розуміти починаючи з пояснень (візуалізації) результатів експерименту і закінчуючи створенням рекламних роликів.

2)*Саморозвиваючий* - КГ повинна обслуговувати свої потреби, розширюючи свої можливості та удосконалюючи їх.

3)*Дослідницький,* в якому інструментарій КГ починає грати роль, яка подібна тій , яку у свій час зіграв мікроскоп.

## 1.1. СКЛАДНІСТЬ АЛГОРИТМІВ

1. **Означення.** *Час, який витрачається алгоритмом, як функція розміру задачі, наз.* ***часовою складністю*** *цього алгоритму. Граничну поведінку цієї складності при збільшені розміру задачі наз.* ***асимптотичною часовою складністю****.* Аналогічно, можна виділити ***об'ємну складність*** та ***асимптотичну об'ємну складність.***

*<u>Оцінки складності.</u>*

$O(f(N))$ - $\{ g(N) \mid \exists\, C>0$ і $N_0>0: |g(N)| \le C f(N), \forall N \ge N_0\}$ (верхні оцінки);

$\Omega(f(N))$ - $\{ g(N) \mid \exists\, C>0$ і $N_0>0: |g(N)| \ge C f(N), \forall N \ge N_0\}$ (нижні оцінки)

$\theta(f(N))$- $\{ g(N) \mid \exists\, C_1, C_2, N_0 >0: C_1 f(N) \le g(N) \le C_2 f(N), \forall N \ge N_0\}$ (ефективні оцінки)

Якщо алгоритм обробляє входи розміру $n$ за час $cn^2$, де $c$-деяка константа, то часова складність цього алгоритму є $O(n^2)$, для усіх n, крім скінченої ( можливо порожньої) множини, невід'ємних значень.

### 1.2. Моделі обчислень

**Означення.** ***Модель обчислень*** *визначає набір допустимих елементарних операцій і вартості цих операцій.*

*Елементарні операції* - це такі операції, для кожної із яких призначається фіксована вартість, хоча вартість неоднакова для різних елементарних операцій.

*<u>Основні моделі обчислень:</u>*

*РАМ- машина з довільним доступом до пам'яті (або, рівно доступна адресна машина );*

*РАСП - машина з довільним доступом до пам'яті і програмою яка зберігається (або, рівно доступна адресна машина із програмою, яка зберігається; машина Тюрінга.*

Модель РАМ (дійсна RAM – Random Access Machine) в кожній комірці пам'яті може зберігати єдине дійсне число та характеризується наступними елементарними операціями, що мають одиничну вартість:

1. Арифметичні операції: +, -, *, /.
2. Операції порівняння двох дійсних чисел: <, <=, =, >=, >.
3. Непряма адресація пам'яті лише з цілочисельними адресами.

При необхідності, використовуються логічні, алгебраїчні та тригонометричні операції.

*Обчислювальна складність РАМ-програм.*

*Складністю в гіршому випадку* - це максимальна міра ефективності даного алгоритму для всіх задач даного розміру.

Якщо за міру складності береться "середня" складність по всім входам даного розміру, то вона наз. *середньою складністю.*

*Часова складність в гіршому випадку* РАМ програми - це функція $f(n)$, рівна найбільшій (по всім входам розміру $n$) із сум часів, витрачених на кожну спрацьовану команду.

*Часова складність в середньому* - це середнє взяте по всім входам розміру $n$ тих же самих сум.

## 1.3. Перетворення (звідність) задач.

**Означення.** Задача **A** може бути *перетворена* в задачу **B** (задача **A** *зводиться* до задачі **B**), якщо:

1. Вхідні дані задачі **A** перетворюються у вхідні дані задачі **B**;
2. Розв'язується задача **B**;
3. Результат розв'язку задачі **B** перетворюється у правильний розв'язок задачі **A**.

Якщо кроки 1 та 3 можна виконати за час $O(r(N))$, де N – розмір задачі **A**, то кажуть, що задача **A** є $r$(N) звідною до **B** і позначають так: $A \infty_{r(N)} B$. Звідність не є симетричним відношенням. Якщо задачі **A** та **B** взаємно перетворюємі, то вони називаються *еквівалентними*.

**Твердження 1.** *(нижні оцінки методом перетворення)*

*Якщо відомо, що задача* ***A*** *вимагає T(N) часу і задача* $A \infty_{r(N)} B$ *(перетворюємо в* ***B****), то* ***B*** *можна розв'язати за час не менший за* $T(N) - O(r(N))$.

**Твердження 2.** *(верхні оцінки методом перетворення)*

*Якщо задачу* ***B*** *можна розв'язати за час T(N) і задача* $A \infty_{r(N)} B$ *(перетворюємо в* ***B****), то* ***A*** *можна розв'язати за час, який не перевищує* $T(N)+O(r(N))$.

## 1.4. Деякі задачі із множини класів звідності задач

### *Задачі класу звідності $\Omega(n\log_2 n)$:*

1. Задача сортування. Задана множина $S$ із $n$ елементів, необхідно її відсортувати: час $\Omega(n\log_2 n)$.

2. Перевірка унікальності елементів: чи є на заданій множині два рівних елементи.
3. Перевірка рівності включення множин.
4. Перевірка перетину та об'єднання множин.
5. Перевірка на значимість: Задана множина з *S* точок на площині: чи усі точки належать опуклій оболонці множини точок.
6. Перевірка ε- близькості: Задана множина з *S*, чи є серед елементів цієї множини два елементи, які знаходяться на відстані ε.
7. Об'єднання та перетин інтервалів.

***<u>Задачі класу звідності Ω(logn):</u>***

1. Перевірка належності елемента $a \in S$ для скінченної множини елементів $S$, ($|S|=n$).
2. Бінарний пошук. Нехай задана скінченна множина елементів $S$, ($|S|=n$). Необхідно знайти найближчий елемент до заданого на $S$.

*Відомий факт.* Якщо деяка задача в d-мірному просторі вимагає *f(n)* часу, то для цієї задачі в розмірності $k>d$ також потрібно як мінімум *f(n)* часу.

## 1.5 Загальні означення

***d - вимірним евклідовим простором*** називається простір d - плексів ($x_1$, $x_2$, ..., $x_d$), що складаються з дійсних чисел $x_i$ i = 1, ..., d з відстанню $\sqrt{\sum_{i=1}^{d} x_i^2}$ .

***Точкою*** p в просторі $E^d$ називається d - плекс ($x_1$, $x_2$, ..., $x_d$).

Нехай дано дві різні точки $q_1, q_2 \in E^d$. *Лінійна комбінація* $aq_1 + (1 - a)q_2$ ($a \in R$) наз. ***прямою в $E^d$***.

Нехай дано дві різні точки $q_1, q_2 \in E^d$. *Лінійна комбінація* $aq_1 + (1 - a)q_2$ ($a \in [0; 1]$) визначає ***опуклу комбінацію*** для $q_1$та $q_2$, яка описує ***прямолінійний відрізок***, що сполучає дві точки: $q_1$ та $q_2$. Цей відрізок позначають так: $\overline{\boldsymbol{q}_1\boldsymbol{q}_2}$ .

***Опуклою оболонкою*** множини точок S, що належать простору $E^d$, називається *границя найменшої опуклої області в $E^d$, що охоплює S.*

***Многокутником*** в просторі $E^2$ називається скінчена множина відрізків, в якому кожний кінець відрізка належить рівно двом відрізкам і жодна підмножина відрізків не має вказану властивість. Ці відрізки називаються **сторонами (ребрами)**, а їх кінці – ***вершинами*** многокутника. Або: *многокутником в $E^d$ називається замкнена n-вершинна ламана.*

*Многокутник* називається ***простим***, якщо жодна пара непослідовних його ребер не має спільних точок. Простий многокутник розбиває площину на дві неперетинаючі області – ***внутрішню*** (скінчену) та ***зовнішню*** (нескінченну).

*Простий многокутник* P називається ***опуклим***, якщо його внутрішня область є опуклою множиною.

*Простий многокутник* називається ***зірковим*** **(зірчатим)**, якщо існує точка z, не зовнішня для P така, що для всіх точок p, що належать P, відрізок $\overline{zp}$ повністю лежить всередині P. *Множина точок* z, яка має вказану властивість, називається ***ядром*** P. Опуклий многокутник співпадає зі своїм ядром.

*Граф* G = (V, E) називається ***планарним,*** якщо його можна покласти на площині без самоперетинів. Прямолінійне укладання ребер планарного графа визначає *розбиття площини*, яке називається ***планарним підрозбиттям*** або ***картою***.

Нехай *v* – число вершин, *e*- число ребер, *f*- число граней (включаючи єдину нескінченну грань) такого розбиття. Ці три параметри зв'язані формулою Ейлера:***v-e+f=2***

***Вправа.*** Якщо відомо, що ступінь к5ожної вершини ≥3, то необхідно довести такі нерівності: v ≤ 2e/3, e ≤ 3v-6e ≤ 3v-6, e ≤ 3f-6, f ≤ 2e/3, v ≤ 2f-4, f ≤ 2v-4.

*Планарне підрозбиття* називається ***тріангуляцією***, якщо всі його скінченні грані є трикутниками. Тріангуляцією скінченої множини точок S називається плоский граф S, який має найбільшу можливу кількість ребер.

**Рекомендована література**


1. Препарата Ф., Шеймос М. Вычислительная геометрия: Введение. Г.: Мир, 1989. – 478 с.
2. Ахо Х., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. М.: Мир, 1979. – 536.
3. Роджерс Д. Алгоритмические основы машинной графики. М.:Мир, 1989.- 504 с.
4. Роджерс Д., Адамс Дж. Математические основы машинной графики. М.: Мир, 2001. 604с.
5. В.М. Терещенко, І.В. Кравченко, А. В. Анісімов. Основні алгоритми обчислювальної геометрії, Київ, 2002р, 81 с.
6. Майкл Ласло. Вычислительная геомтрия и компьютерная графика на С++. М.:Бином, 1997.-301 с.
7. Goodman J.E., O'Rourke J. Handbook of Discrete and Computational Geometry. - N.Y.: Chapman and Hall/CRC Press, 2004. – 1497 p.
8. Mark de Berg, Otfried Cheong, Marc van Kreveld, Mark Overmars. Computational Geometry:Algorithms and Applications.Berlin Heidelberg: Springer-Verlag, 2008. – 386 p.