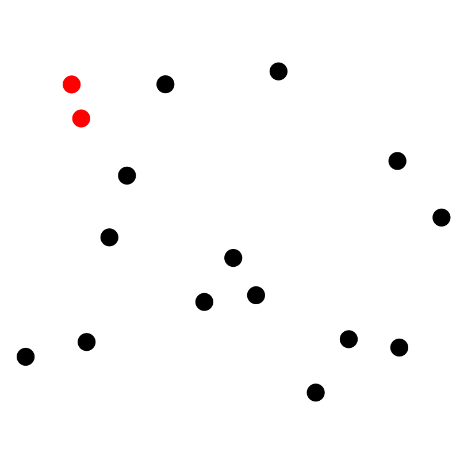
Машир Анна, КМ-93

**Завдання пошуку найближчих точок за допомогою використання методу грубої сили**

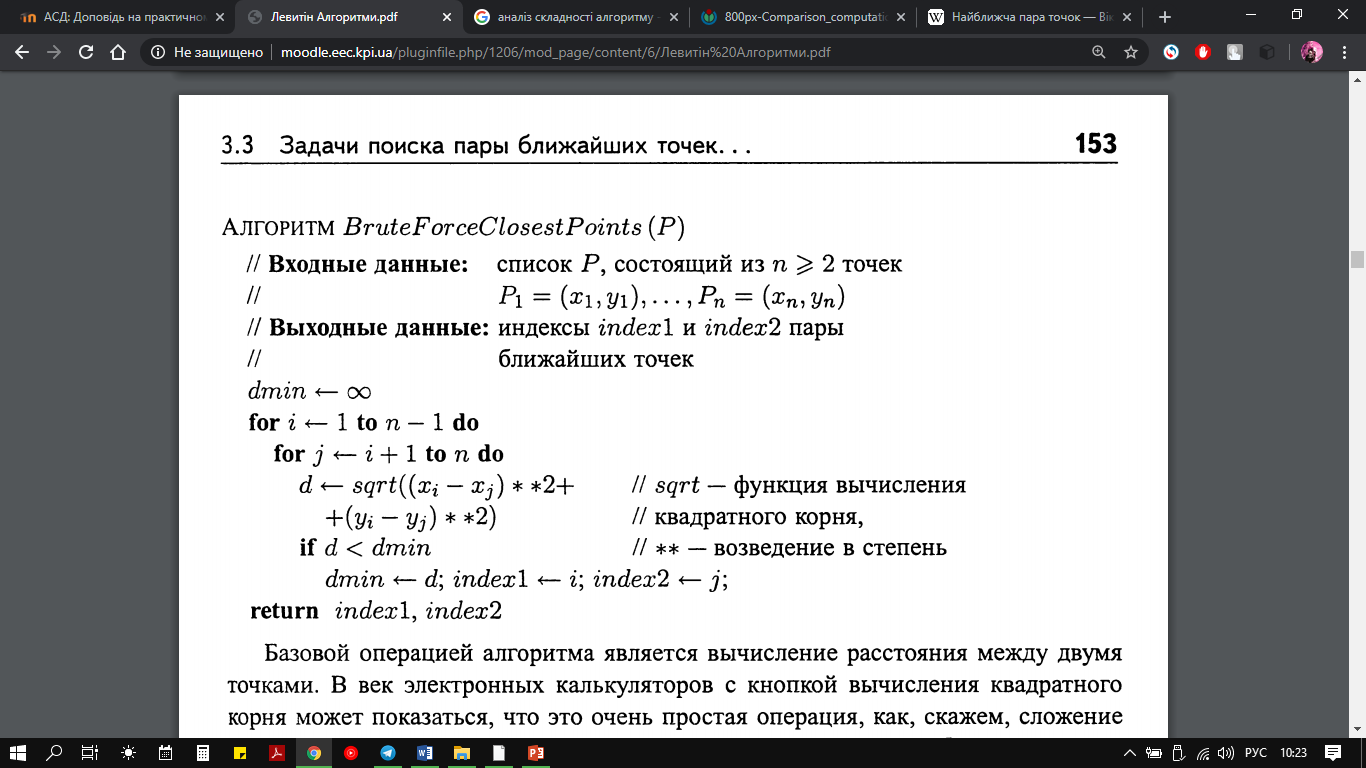
Сенс в тому, щоб з множини з n точок відшукати дві, які розташовані найближче.



Для простоти прикладу будемо вважати що у нас двовимірний простір. Точки задаються стандартним способом за допомогою декартових координат (х, у). Відстань між двома точками являє собою стандартну відстань геометрії Евкліда: , ) = .

Щоб уникнути повторного обчислення відстані між одними і тими ж точками розглядатимемо пари точок , ), для яких i <j.

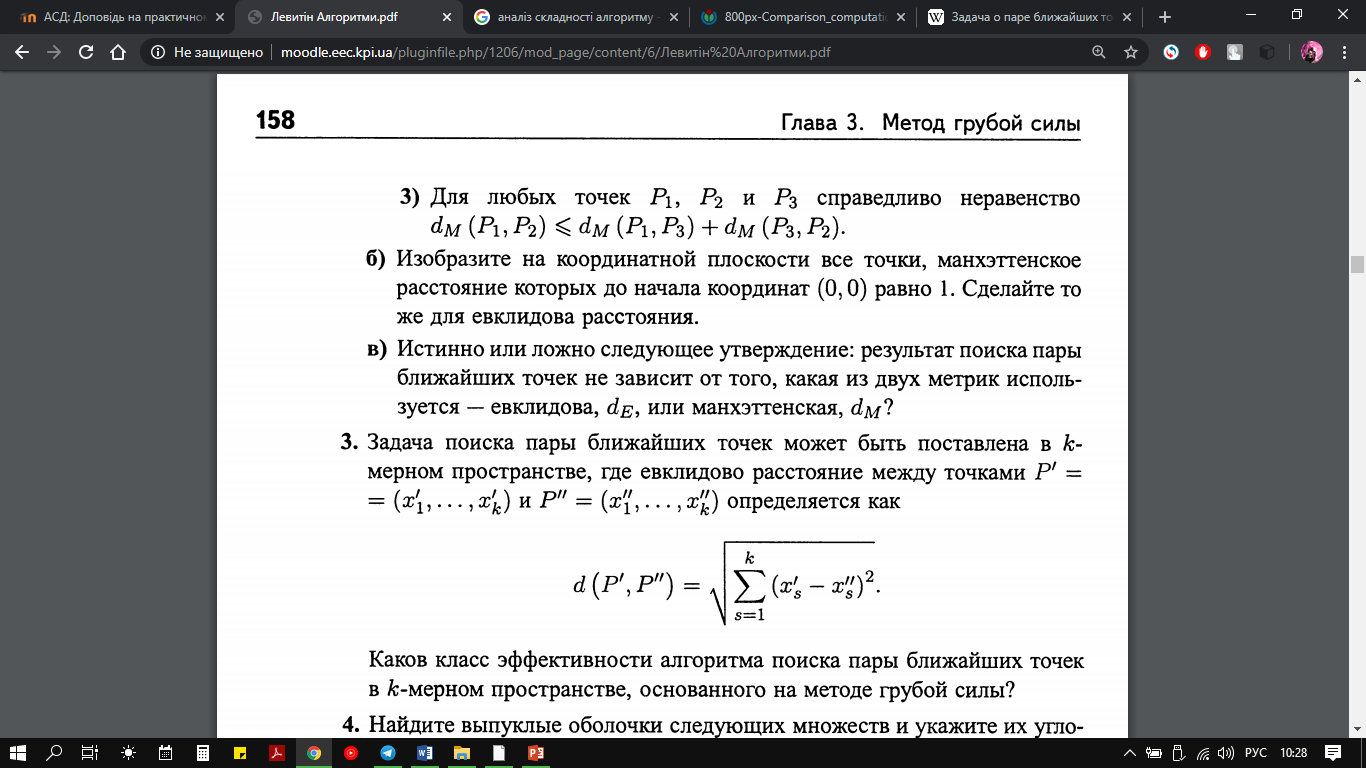
Псевдокод:



Аналіз складності алгоритму:

Пара найближчих точок може бути обчислена за час O(n2) шляхом виконання повного перебору. Щоб це зробити, можна обчислити відстань між усіма парами точок, потім вибрати пару з найменшою відстанню.

Вправа:



Код реалізовано на мові Python:

import math  
def find\_min\_distance():  
 min\_distance = math.inf  
 for i in range(k-1):  
 for j in range(i+1, k):  
 sub = 0  
 for a in range(n):  
 sub += (arr[i][a] - arr[i+1][a])\*\*2  
  
 distance = math.sqrt(sub)  
 if distance < min\_distance:  
 min\_distance = distance  
 points = [i, i+1]  
 print('\nМінімальна відстань: ', min\_distance, "\nМіж точками номер ", points[0], " та ", points[1])  
  
n = int(input("Введіть розмірність простору "))  
k = int(input("Введіть кількість точок "))  
arr = []  
for i in range(k):  
 cords = []  
 while len(cords) != n:  
 print("Введіть координати точки ", i+1, " через пробіл:")  
 cords = input().split(" ")  
 cords\_int = [int(i) for i in cords]  
 arr.append(cords\_int)  
print(arr)  
find\_min\_distance()

Скріншоти тестування:

