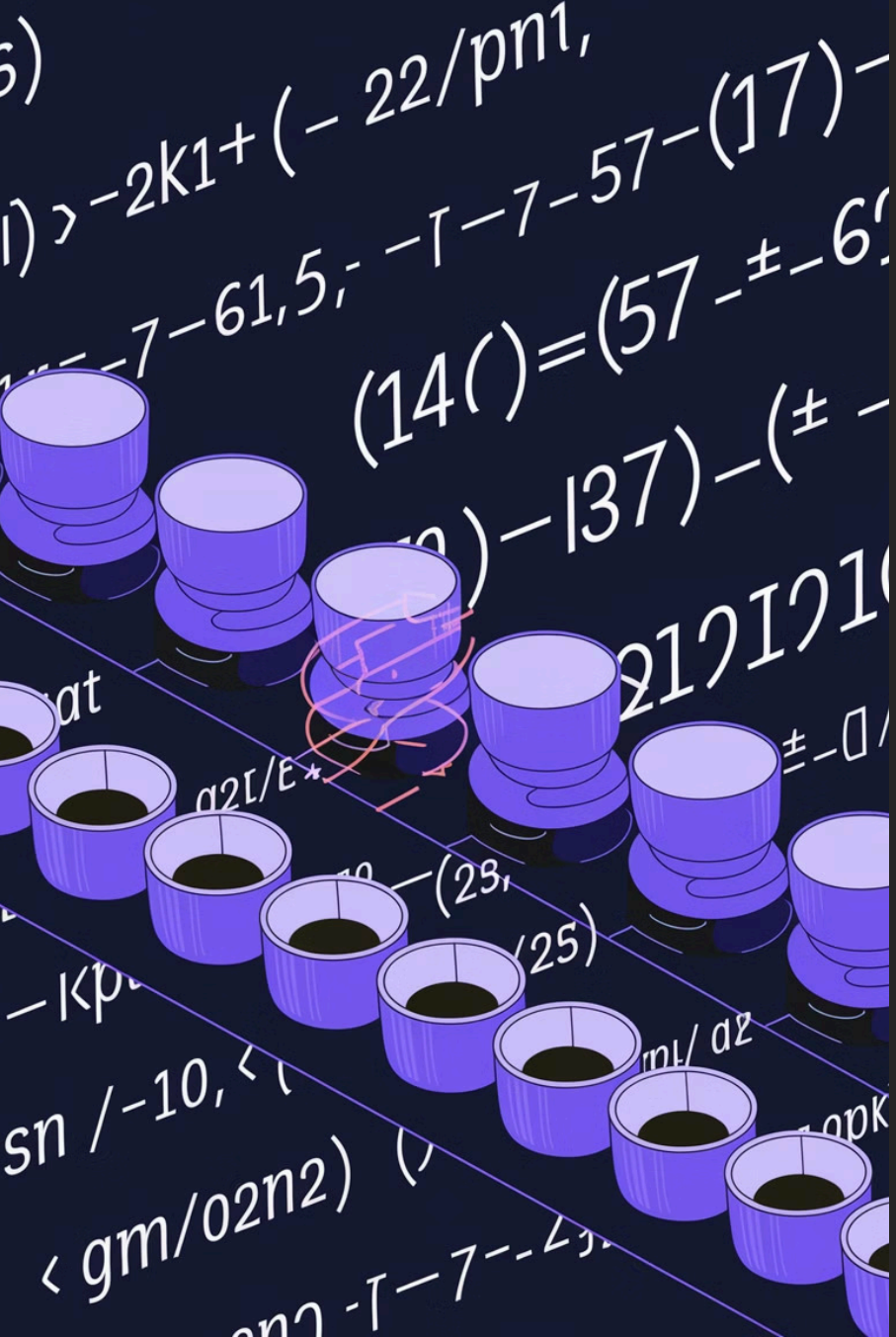
Знаходимо найближче кафе: програмування для початківців

Вітаємо на практичному занятті з програмування! Сьогодні ми навчимося знаходити найближче кафе за допомогою алгоритмів та математики.

by o d



Що ми будемо робити?

1 **Вивчення масивів і циклів**

Навчимося працювати з даними, перебирати їх та аналізувати.

2 **Застосування математики**

Використаємо формулу евклідової відстані для знаходження найближчої точки.

3 **Розв'язання практичної задачі**

Допоможемо туристу знайти найближче кафе в незнайомому місті.

Уявімо ситуацію

Турист

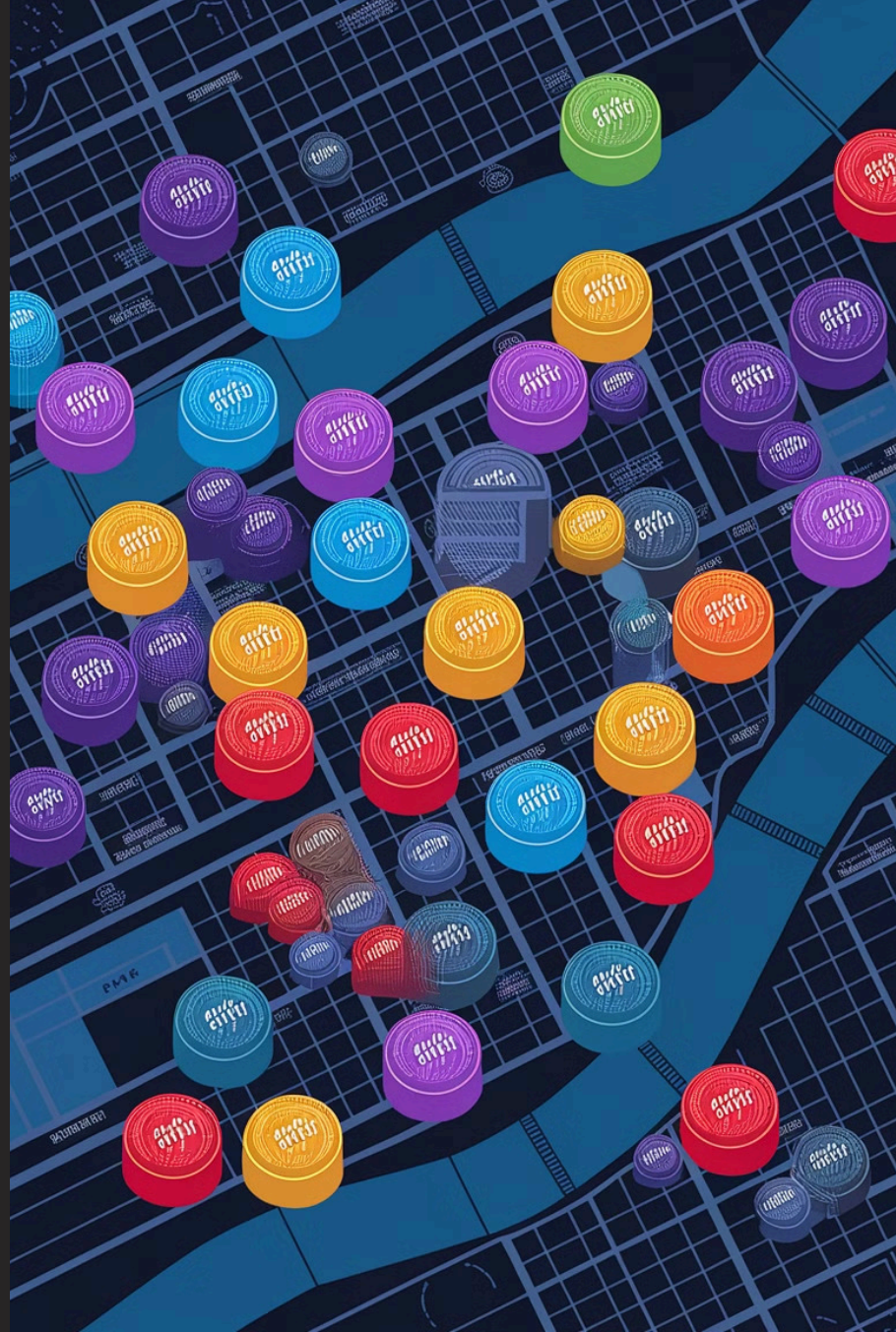
Ви - турист у незнайомому місті. Хочете знайти найближче кафе для перепочинку.

Карта

У вас є карта з координатами різних кафе та ваше поточне місцезнаходження.

Завдання

Потрібно написати програму для визначення найближчого кафе до вашої позиції.



Вхідні дані

Масив кафе

Маємо список кафе з їхніми координатами (x, y) та назвами.

```
const cafes = [
  { name: "Latte Heaven", x: 3, y: 7 },
  { name: "Pizza Corner", x: 5, y: 2 },
  { name: "Sushi Point", x: 9, y: 6 },
  { name: "Burger Land", x: 2, y: 3 },
  { name: "Coffee Time", x: 6, y: 8 }
];
```

Позиція туриста

Знаємо, де саме знаходиться наш турист.

```
const tourist = {
  x: 4,
  y: 5
};
```

Що таке евклідова відстань?

Теорема Піфагора

Евклідова відстань базується на теоремі Піфагора, яку ми вивчали в школі.

Формула

Відстань = √((x2-x1)² + (y2-y1)²), де (x1,y1) і (x2,y2) - координати точок.

Застосування

Ця формула допоможе нам визначити найкоротшу відстань між туристом і кафе.

Алгоритм пошуку найближчого кафе

1

Ініціалізація

Створюємо змінні для зберігання мінімальної відстані та найближчого кафе.

2

Перебір усіх кафе

Проходимо циклом по всіх кафе з нашого масиву.

3

Обчислення відстані

Для кожного кафе обчислюємо евклідову відстань до туриста.

4

Порівняння

Якщо знайдена відстань менша за мінімальну, оновлюємо найближче кафе.

Реалізація функції пошуку

```
function findNearestCafe(cafes, tourist) {
  let minDistance = Infinity;
  let nearestCafe = null;

  for (let i = 0; i < cafes.length; i++) {
    let cafe = cafes[i];
    let distance = Math.sqrt(
      Math.pow(cafe.x - tourist.x, 2) +
      Math.pow(cafe.y - tourist.y, 2)
    );

    if (distance < minDistance) {
      minDistance = distance;
      nearestCafe = cafe;
    }
  }

  return nearestCafe;
}
```

Розбір коду

1

Ініціалізація змінних

minDistance = Infinity: Спочатку встановлюємо відстань як нескінченність. nearestCafe = null: Найближче кафе поки не визначене.

2

Цикл перебору

Використовуємо цикл for для перебору кожного кафе в масиві cafes.

3

Обчислення відстані

Застосовуємо формулу евклідової відстані для кожного кафе.

4

Оновлення результату

Якщо знайдена відстань менша за поточну мінімальну, оновлюємо значення змінних.

Виклик функції та виведення результату

Виклик функції

```
const nearest = findNearestCafe(cafes, tourist);
```

Виведення результату

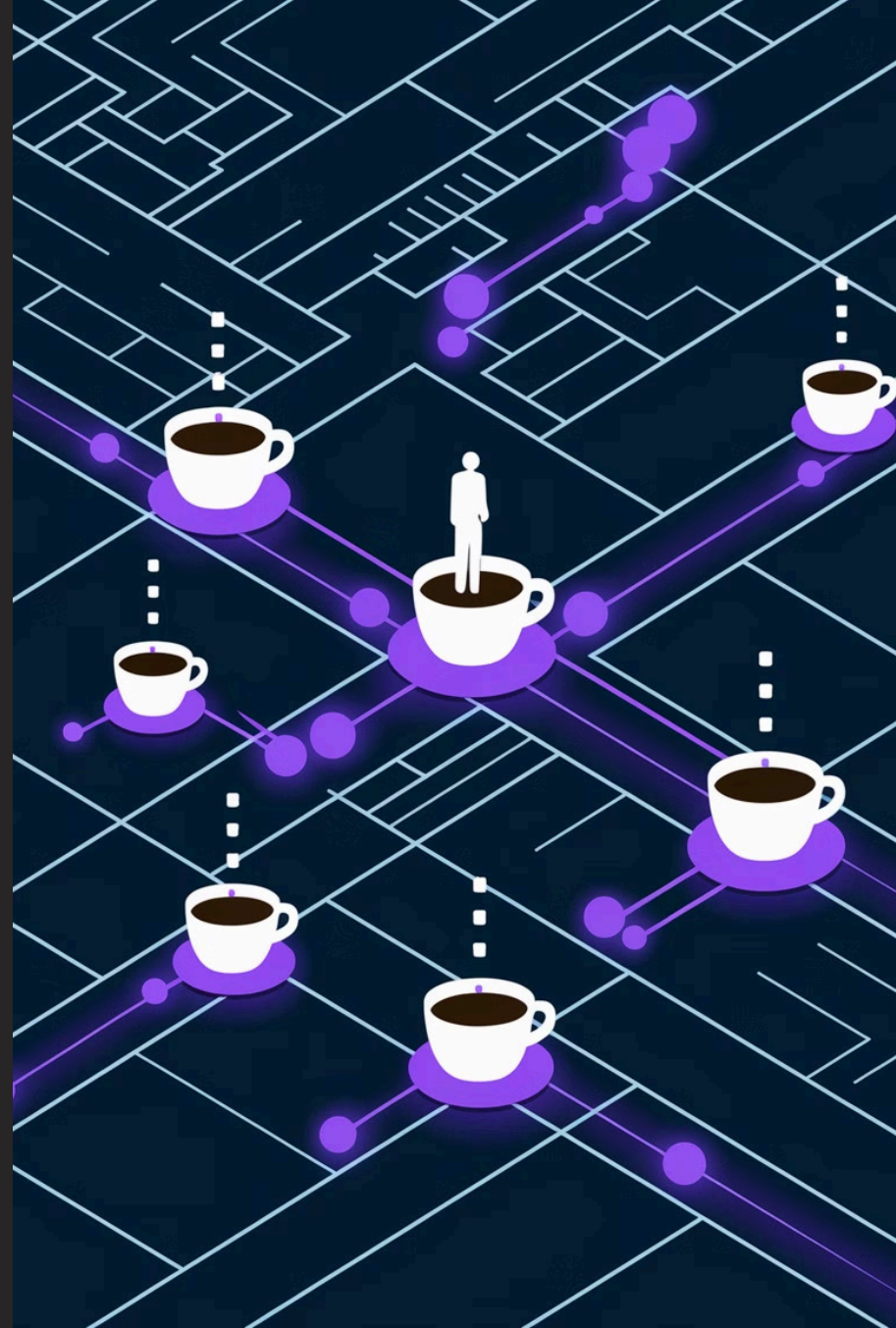
```
console.log(`Найближче кафе: ${nearest.name} (${nearest.x}, ${nearest.y})`);
```

Очікуваний результат

Функція поверне об'єкт з найближчим кафе та його координатами.

Візуалізація на координатній площині

На цій візуалізації ми бачимо розташування туриста (4,5) та всіх кафе на координатній площині. Пунктирними лініями показані відстані від туриста до кожного кафе.



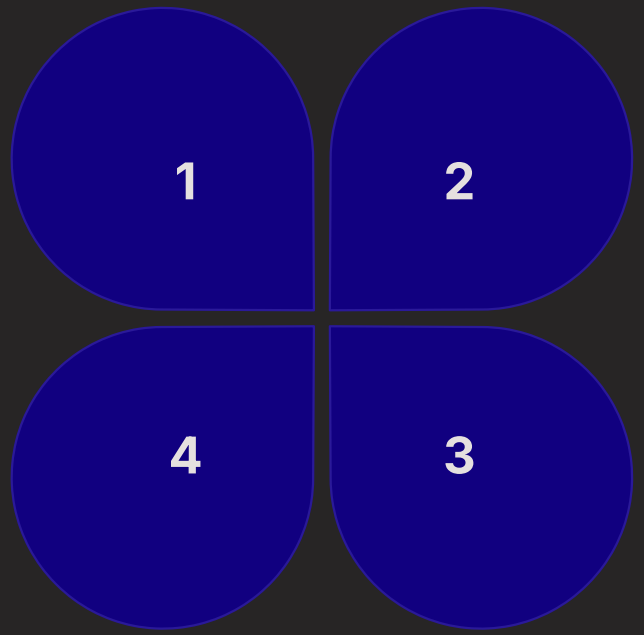
Метод k-найближчих сусідів (k-NN)

Класифікація

Метод використовується для класифікації об'єктів за близькістю до інших об'єктів.

Наш приклад

Ми використовуємо k=1, тобто шукаємо тільки одне найближче кафе.



Параметр k

Вказує кількість найближчих сусідів, які ми хочемо знайти.

Застосування

Рекомендаційні системи, розпізнавання образів, машинне навчання.

Розширення задачі: топ-2 найближчих кафе

Зміна алгоритму

Замість одного найближчого кафе шукаємо два.

1

2

Структура даних

Використовуємо масив для зберігання топ-2 кафе.

3

Сортування

Сортуємо кафе за відстанню і вибираємо перші два.

Ускладнення: вибір кафе за рейтингом

Дані про рейтинг

Додаємо рейтинг до об'єктів кафе

1

2

Розрахунок метрики

Комбінуємо відстань та рейтинг

3

Оптимальний вибір

Обираємо найкраще кафе з найближчих

Врахування рейтингу дозволяє знайти баланс між відстанню до кафе та якістю обслуговування.

Алгоритм з урахуванням рейтингу

Можна створити зважену метрику, яка враховує і відстань, і рейтинг кафе. Наприклад: score = distance * (5 - rating).

Підсумки та практичне застосування

1

Масиви і цикли

Ми навчилися працювати з масивами об'єктів і проходити циклом по елементах.

2

Евклідова відстань

Застосували математичну формулу для знаходження відстані між точками.

3

k-NN

Освоїли основи алгоритму k-найближчих сусідів на простому прикладі.

5+

Розширення

Познайомилися з можливими ускладненнями задачі та методами їх вирішення.