

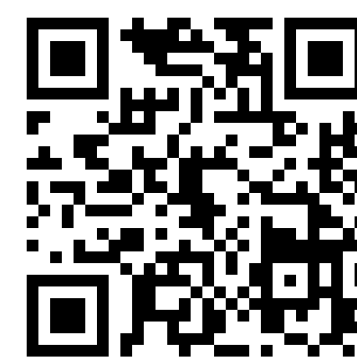


Busan science high school

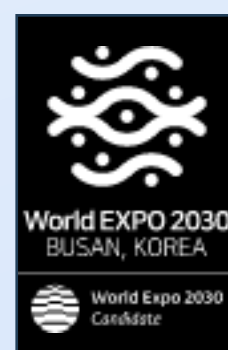
2023 Ocean ICT Festival

2023 BOIF

A
03

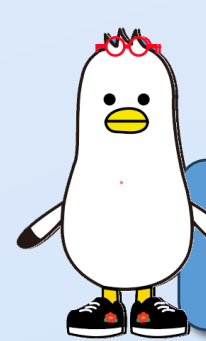


Youtube 영상 QR



ACO algorithm을 이용해

2030 엑스포를 위한 해상 도시 계획의 교통문제 보완하기

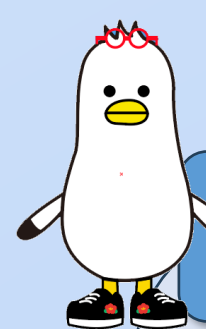


탐구동기

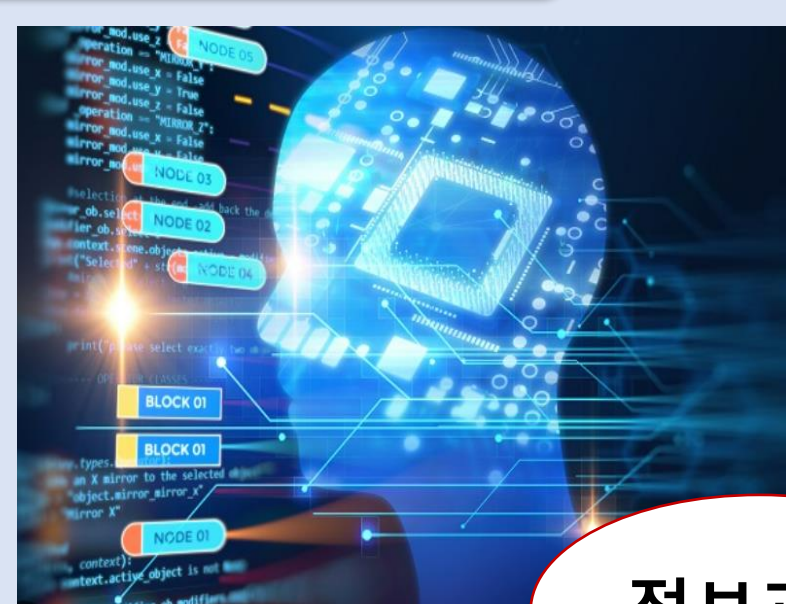
개미 크루 2311 김민수 / 2312 김채준

2030 부산 월드 엑스포 개최 유치를 위해
정부에서는 많은 노력들을 기울이고 있다.
그 중 정부에서 계획 중인 정책 하나는
부산에 관광 해상도시를 건설하는 것이다.

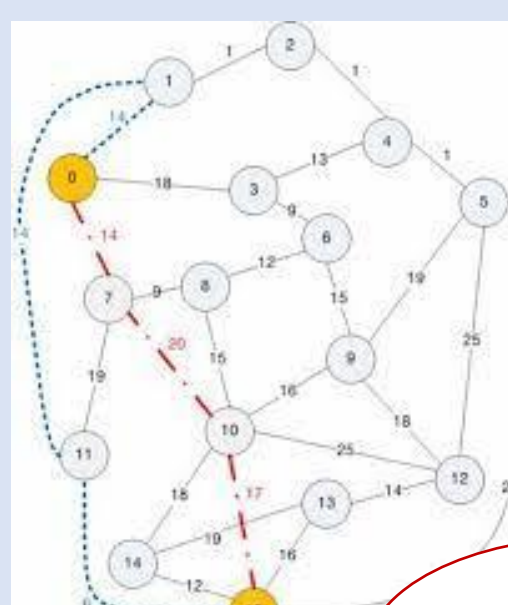
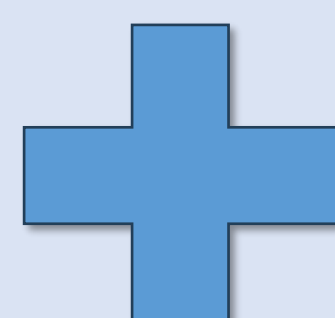
부산의 많은 섬들을 이용해 관광 해상도시를 건설한다면
섬들 사이의 이동이 효율적이어야 한다.
이에 우리는 섬들간의 이동 중
최적의 경로를 찾는 프로그램을 개발하였다.



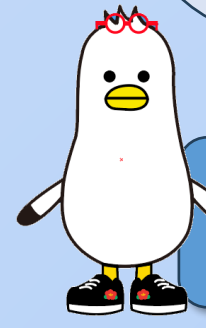
융합 분야



정보과학



생명과학

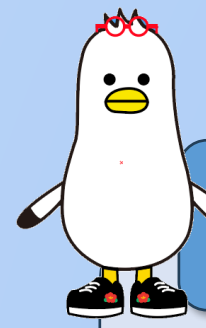


작품설명

부산에 위치한 약 50개 섬들의 위도와 경도를 적절한 노드로 변경한다.
변경한 섬의 좌표들을 바탕으로 최적의 경로를 찾는다.
사용자의 임의로 최적의 경로를 찾을 개미의 군집을 직접 설정할 수 있다.
개미의 개체 수 등을 임의로 변경함으로써 프로그램이 최적의 경로를 찾고
시각화 시킴으로써 한 눈에 알아 볼 수 있도록 한다.



BUSAN
is Ready!



코드설명

```
import operator
import matplotlib.pyplot as plt

def plot(points, path: list):
    x = []
    y = []
    for point in points:
        x.append(point[0])
        y.append(point[1])
    # noinspection PyUnresolvedReferences
    y = list(map(operator.sub, [max(y) for i in range(len(points))], y))
    plt.plot(x, y, 'co')

    for i in range(1, len(path)):
        j = path[i - 1]
        k = path[i]
        # noinspection PyUnresolvedReferences
        plt.arrow(x[i], y[i], x[j] - x[i], y[j] - y[i], color='r', length_includes_head=True)

    # noinspection PyTypeChecker
    plt.xlim(0, max(x) + 1.1)
    # noinspection PyTypeChecker
    plt.ylim(0, max(y) + 1.1)
    plt.show()
```

그래프에 점을 찍고
경로를 시각화 시켜주는
코드

```
import math

def distance(city1: dict, city2: dict):
    return math.sqrt((city1['x'] - city2['x']) ** 2 + (city1['y'] - city2['y']) ** 2)

def main(_signal, file, number, generations, alpha, beta, rho, q, strategy):
    cities = []
    points = []
    with open(file) as f:
        for line in f.readlines():
            city = line.split(' ')
            cities.append(dict(index=int(city[0]), x=int(city[1]), y=int(city[2])))
            points.append((int(city[1]), int(city[2])))
    cost_matrix = []
    rank = len(cities)
    for i in range(rank):
        row = []
        for j in range(rank):
            row.append(distance(cities[i], cities[j]))
        cost_matrix.append(row)
    aco = ACO(number, generations, alpha, beta, rho, q, strategy)
    graph = Graph(cost_matrix, rank)
    path, cost = aco.solve(graph)
    if _signal==1:
        plot(points, path)
        print('경로의 길이 cost : {}'.format(cost))
        print('경로 path: {}'.format(path))
    if _signal==0:
        print(cost)
```

노드 사이의 거리를
계산하고 개미 알고리즘
으로 최적의 경로를 계
산해주는 코드

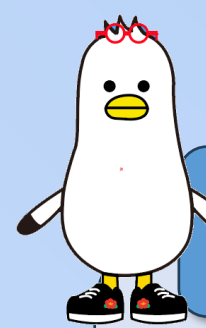
```
def result(signal, ant_number, genera):
    if __name__ == '__main__':
        main(signal, 'city.txt', ant_number, genera, 1.0, 10.0, 0.5, 10, 2)

from time import sleep
def show():
    print("LOADING")
    sleep(1)
    print("1")
    sleep(1)
    print("2")
    sleep(1)
    print("3")
    sleep(1)
    print("4")
    sleep(1)
```

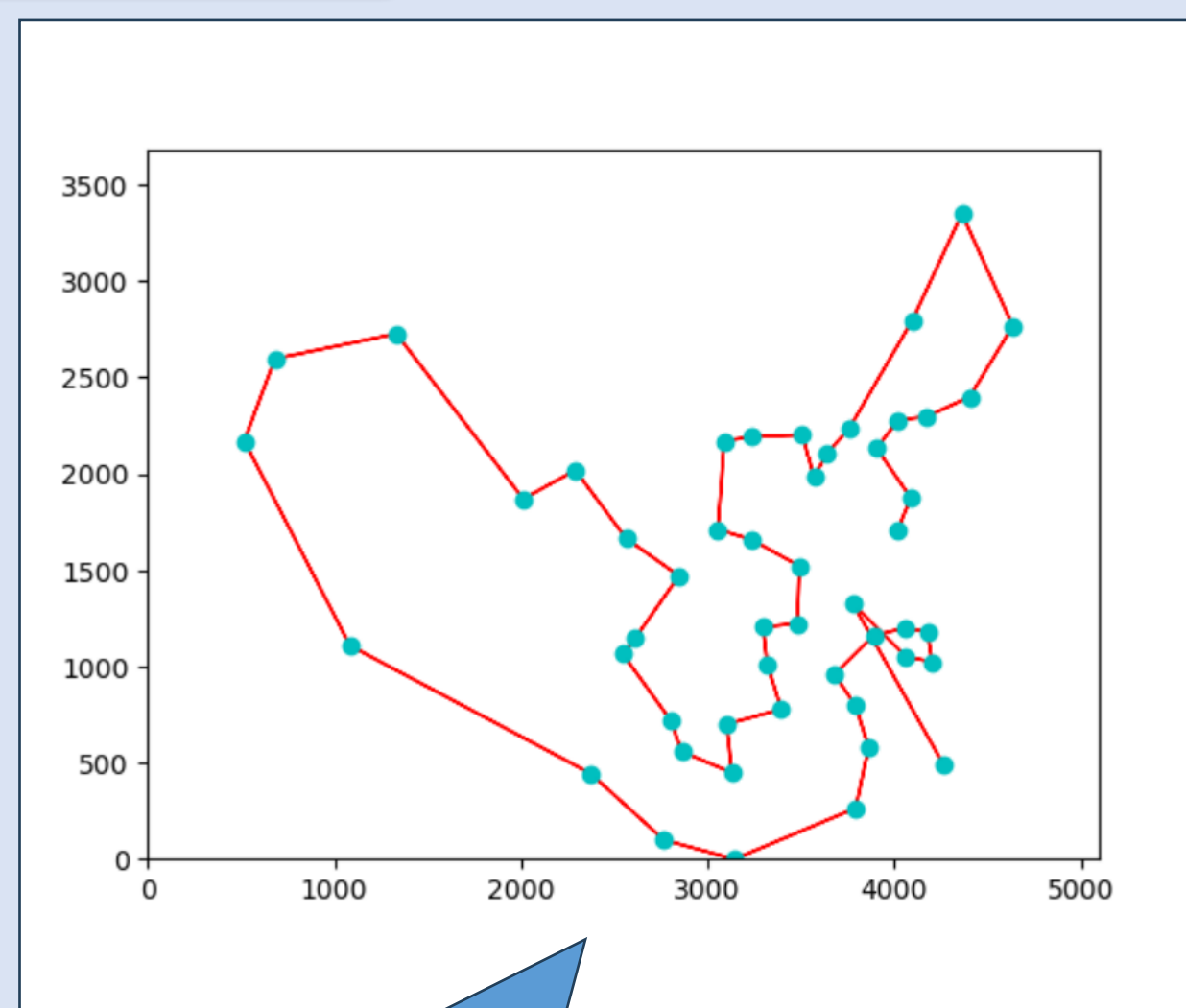
사용자가 원하는 결과를
출력해주고 이용 시
진행의 원활을 돕는 코드

```
import matplotlib.pyplot as plt
def extract_column_numbers(file_path, column_index):
    a = []
    with open(file_path, 'r') as file:
        for line in file:
            columns = line.strip().split()
            if len(columns) > column_index:
                a.append(int(columns[column_index]))
    return a
```

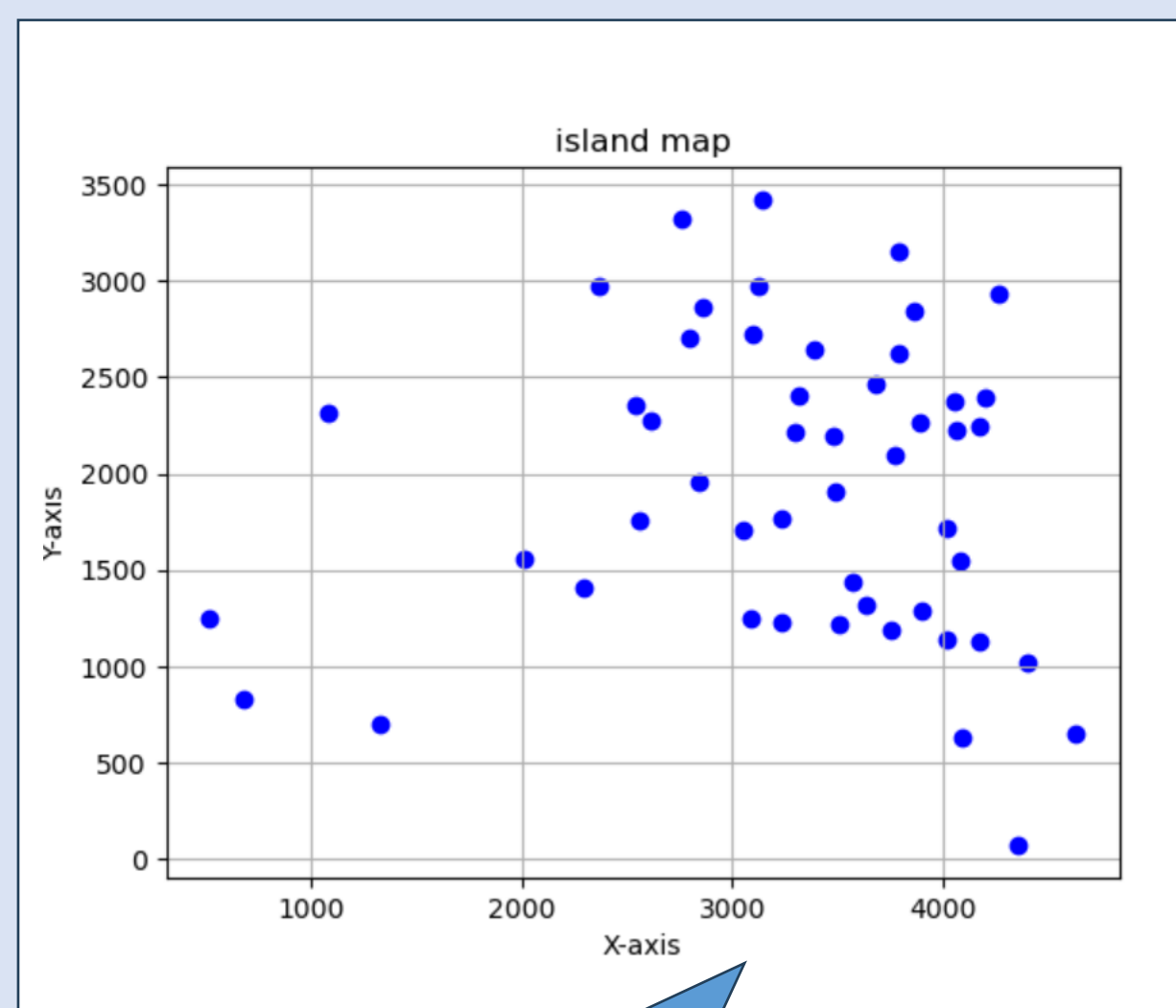
섬 좌표가 있는 파일
에서 좌표를 가져와
그래프로 도시화
시키는 코드



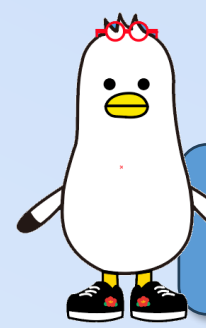
결과화면



사용자가 정한 개미 개체수와
세대의 수를 바탕으로
최적의 경로를 찾아서
그래프로 시각화 시킨 화면



부산 지역의 섬 좌표들을
그래프에 도식화
시킨 후
출력 시킨 화면



기대효과

국제 박람회(EXPO)를 유치하는 것 만으로도 국가에서는 매우 큰 경제 효과를
가져와 줍니다.
그렇기에 많은 나라에서 EXPO 유치를 위해 힘을 씁니다.
부산 해상 도시 계획을 성공 실행 시킨다면 EXPO 유치에 꽤나 큰 기여를 할 수 있습니다.
개미 알고리즘을 이용하여 해상 교통 문제를 해결한다면
'자연을 모방하였다는 점', '환경 보호에 크게 기여 했다는 점'에서
다른 나라로부터 좋은 이미지를 심는데 일조할 것입니다.
부산 세계 엑스포 유치를 응원합니다.

BUSAN
is Ready!

