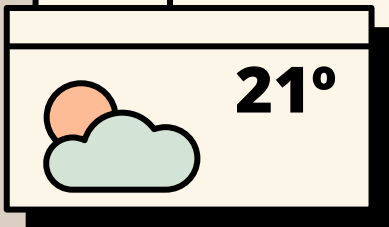
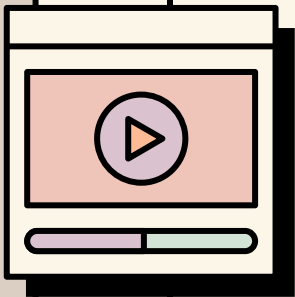


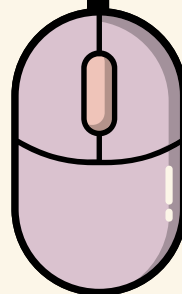
Final Project Updates



>>>>

--Yinuo Xu, Junyi Pan

~~~~~  
.....



# Methodology

## Different Algorithms

**Heuristic Algorithms:** Use heuristic algorithms such as genetic algorithms to quickly generate near-optimal solutions.

**Local Search:** 2-opt or 3-opt to improve existing routes by iteratively swapping edges to reduce total distance

.....

>>>>>



# What we finished .....

- Heuristic Algorithm
  - Genetic Algorithm
- Idea
  - Chromosomes with a number of genes
    - encoded decision
    - Randomly generated tracks with random routes
  - Crossover
    - Swap parent genes
    - Alternating edges crossover

p1 (5 1 7 8 4 9 6 2 3)

p2 (3 6 2 5 1 9 8 4 7)

c (5 1 9 6 2 3 ...)
  - Mutation



»»»»

# What we plan for next phrase? ~~~~

## Parallel Small task

- Parallelize by using `tbb::parallel_for` to generate a subset of the total number of chromosomes concurrently
- Apply parallelism to the cost calculation which is independent of others.
- Make sure each subset of cost can be processed concurrently by different threads or tasks, like `tbb::parallel_reduce` or `_for`

## Parallel Data Structure

- Utilize concurrent data structures from Intel TBB, like `tbb::concurrent_vector`
- Store or share solution and subsets of chromosomes accessed by multiple threads.

# Speed UP - OSRM

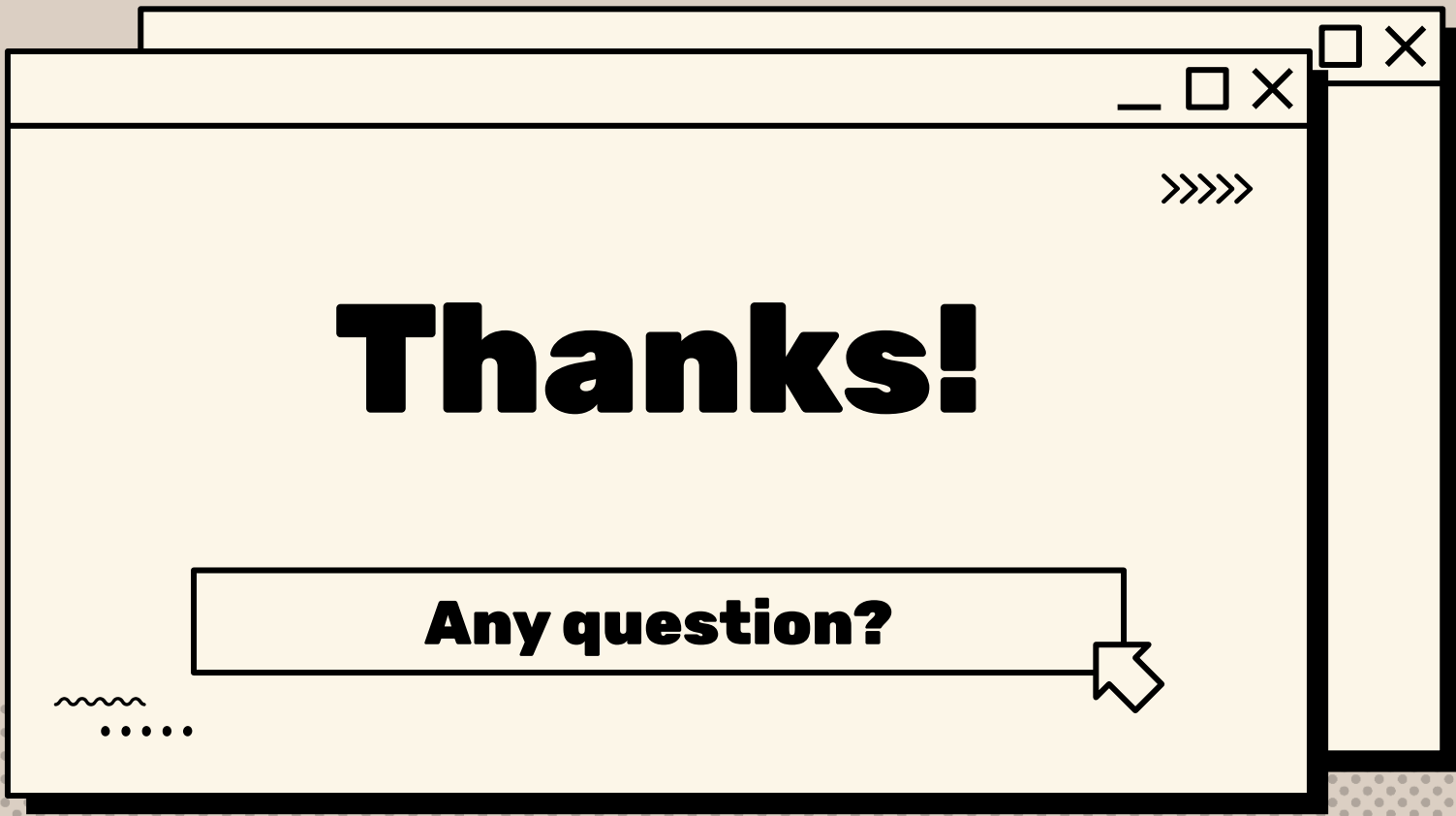
- The Open Source Routing Machine (OSRM) is a high-performance routing engine designed to efficiently compute shortest paths and travel distances between locations.
- It offers an API for querying routing information, including distances, travel times, and optimized routes. OSRM is open-source, actively maintained, and optimized for performance.

.....

>>>>>



- We could use in the crossover step to select a subset of genes from the parents for recombination.
- It is efficient to deal with a large number of locations and optimize the vehicle routes by using preprocessed road network data
- Integration of ORSM could reduce the computational overhead and faster the execution times.





# Reference

[https://www.researchgate.net/publication/268043232\\_Comparison\\_of\\_eight\\_evolutionary\\_crossover\\_operators\\_for\\_the\\_vehicle\\_routing\\_problem](https://www.researchgate.net/publication/268043232_Comparison_of_eight_evolutionary_crossover_operators_for_the_vehicle_routing_problem)

[https://www.tutorialspoint.com/genetic\\_algorithms/genetic\\_algorithms\\_mutation.htm](https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_mutation.htm)

[https://www.youtube.com/watch?v=VOE0U3he8rU&t=314s&ab\\_channel=NoorjaxConsulting](https://www.youtube.com/watch?v=VOE0U3he8rU&t=314s&ab_channel=NoorjaxConsulting)