CS164_LBA

December 15, 2024

1 Introduction

Many freshmen are planning to come to Seoul from San Francisco. Good time!

We help organize your memorable last-minute trip to SF while giving you a travel plan for South Korea upon your arrival at ICN International Airport. For this, we have developed a web service that provides an optimal trip plan for both cities, available here (source code).

Follow the website instructions and specify where you want to start the trip in San Francisco. Then depending on the choice of transportation (driving, public transit, walking), it will show you the optimal trip within San Francisco. You could follow the same process for Seoul.

2 Destinations

We chose the following destinations in San Francisco:

1. Chinatown: Chinatown has historic significance to the city as well as lots of good food.
2. Exploratorium: The Exploratirum has a lot of scientific exhibitions that users can interact with. Students can visit for free once a year.
3. Pier 39: Pier 39 is a fun and busy spot with lots of shops selling San Francisco memorabilia. The pier also has views of the California sea lions. It's free to see the sea lions.
4. Painted Ladies: The San Francisco Painted Ladies are the most popular of the painted ladies. The
5. California Academy of Sciences: The California Academy of Sciences has several appealing sites, such as an aquarium, a planetarium and a natural history museum all in one roof. The destination has an annual free pass for residents of San Francisco
6. Lombard Street: Lombard street is also a famous tourist spot. People come to see the 'most twisted road in the world'.The street is also free to visit.
7. Palace of Fine Art: The Palace of Fine Art is one of the landmarks of San Francisco with views of the Golden Gate bridge. The palace is also free to visit.
8. City Hall: San Francisco City Hall is for people interested in civic aspects of the city. It is also close to the Civic Center and the Asian Art Museum. Students can get free tours of the building.

The starting point for the trip is up to the user as long as it's within San Francisco. It will be factored in the trip plan (9 locations in total).

We chose the following destinations in Seoul

1. Dongdaemun Design Plaza (DDP): The DDP is a major urban development landmark in Seoul. It has a design museum, exhibition halls, and a park.

2. Convention & Exhibition Center (COEX) is a large convention center in Seoul which hosts various events, conferences, and exhibitions. It has a mall, a movie theater, and an aquarium. The COEX is free to visit.
3. Namsan Tower (officially YTN Seoul Tower) is communication and observation tower in central Seoul. It is the second-highest point in Seoul with paranomic views of the city.
4. Seoul Station is the main railway station in Seoul. It is the terminus for the KTX and KTX-Sancheon high-speed rail services. The station is also a major hub for the Seoul Metropolitan Subway.
5. Jamsil Lotteer is a major shopping mall in Seoul. It has a department store, a movie theater, an aquarium, and an ice rink. The mall is free to visit.

Similarly, the starting point for the trip is up to the user as long as it's within Seoul. It will be factored in the trip plan (6 locations in total). In the appendix we demo with Minerva residence as the starting point.

3 TSP Formulation

The problem can be divided into two parts: the Traveling Salesman Problem (TSP) for 9 locations in San Francisco and the TSP for 5 locations in Seoul. The goal is to find the optimal tour that visits each location exactly once and returns to the starting point while minimizing the total travel time.

We formulate this as an integer programming problem and use Miller–Tucker–Zemlin (MTZ) condition to eliminate subtours.

$$\min \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} c_{ij} x_{ij}$$

subject to

$$\begin{aligned}
& x_{ij} = 0 \text{ or } 1, \quad \forall i, j = 1, \dots, N \\
& \sum_{i=1}^{N} x_{ij} = 1, \quad \forall j = 1, \dots, N \\
& \sum_{j=1}^{N} x_{ij} = 1, \quad \forall i = 1, \dots, N \\
& u_i \geq 0, \quad i = 1, \dots, N \\
& u_i - u_j + N x_{ij} \leq N - 1, \quad i \neq j, \ i, j = 2, \dots, N
\end{aligned}$$

where:

- x_{ij} is a binary variable that is equal to 1 if the tour goes from location i to location j, and 0 otherwise,
- c_{ij} is the distance of traveling from location i to location j (we chose minutes as metric),
- u_i is an auxiliary variable representing the position of location i in the tour.
- $N = 9$ is the number of locations.

Out of 5 constraints, the first three are the standard constraints for the TSP problem. The last two constraints are the Miller-Tucker-Zemlin (MTZ) constraints to eliminate subtours.

- (1) indicates that the decision variable x_{ij} is a binary variable. $x_{ij} = 1$ if the tour includes a direct path from location i to location j; 0 otherwise.
- (2) ensures that we depart from each location exactly once
- (3) ensures that we arrive at each location exactly once.
- (4) ensures that the auxiliary variable u_i is non-negative, because the position of location i in the tour must be between 1 and N.
  (5) ensures consistent ordering of the locations in the tour. If $x_{ij} = 1$, then $u_i - u_j = 1$. This constraint ensures that subtours are eliminated.

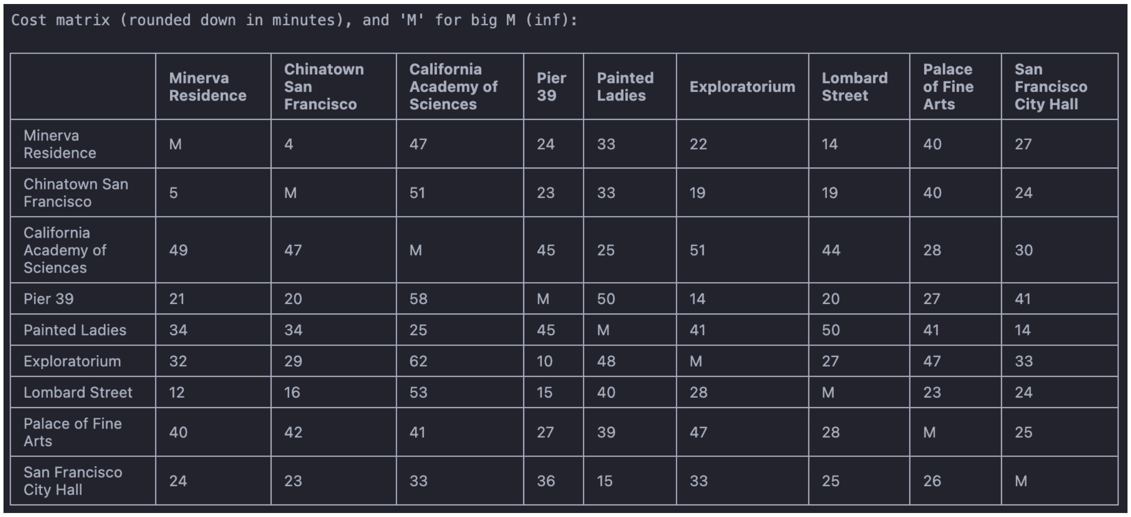
4 Optimal Solution

5 Cost Matrix

We obtained the time traveled between pairs of locations by using Google Maps's API. The API allows us to get the time by different modes of transportation: driving, public transit and walking. Google Maps is also widely used by students in San Francisco to get directions, hence our estimates reflect the distances the students would experience when going to those destinations.

The distance matrix is obtained using the function `get_distance_matrix` in the Appendix.

Below is a sample distance matrix for the locations in San Francisco.



Cost matrix (rounded down in minutes), and 'M' for big M (inf):

	Minerva Residence	Chinatown San Francisco	California Academy of Sciences	Pier 39	Painted Ladies	Exploratorium	Lombard Street	Palace of Fine Arts	San Francisco City Hall
Minerva Residence	M	4	47	24	33	22	14	40	27
Chinatown San Francisco	5	M	51	23	33	19	19	40	24
California Academy of Sciences	49	47	M	45	25	51	44	28	30
Pier 39	21	20	58	M	50	14	20	27	41
Painted Ladies	34	34	25	45	M	41	50	41	14
Exploratorium	32	29	62	10	48	M	27	47	33
Lombard Street	12	16	53	15	40	28	M	23	24
Palace of Fine Arts	40	42	41	27	39	47	28	M	25
San Francisco City Hall	24	23	33	36	15	33	25	26	M

We use the distance matrix and the TSP formulation in `cvxpy` to solve the TSP problem. The function `solve_tsp` in the Appendix solves the TSP problem and returns the optimal tour and the total travel time. In Appendix, we show the optimal tour for the three modes of transportation in San Francisco, and for transit in Seoul.

5.1 Limitations and Future Work

Google Map API fetch time traveled in real-time (factoring in traffic conditions). It works well for SF locations but not for Seoul locations (except for transit mode). Future work could consider using Naver Maps API for Seoul locations.

The current implementation does not consider the time spent at each location.

6 Division of Labor

Woo took pictures for Seoul locations and Ife took pictures for San Francisco locations (for Chiffon and Okomba). All members discussed and agreed on things together.

- Chiffon: Formulated the TSP problem and wrote the optimal solution section.
- Ife: Wrote the Appendix code and the wesbite.
- Okomba: Assisted Ife with code and worked on
- Woo: Picked the theme and wrote skeleton of the report, and added Seoul coordinates to the distance matrix.

The work has been distributed with each other and everyone has chipped in to serve their best with their skills.

7 AI Statement

We used Github Copilot to speed up development of the website and the TSP code.

8 Appendix

[3]:

```python
import os
import time
import googlemaps
import numpy as np
from dotenv import load_dotenv
import cvxpy as cp
from datetime import datetime

# Load the .env file
load_dotenv()

# Initialize Google Maps API
API_KEY = os.getenv("API_KEY")
gmaps = googlemaps.Client(key=API_KEY)


def get_distance_matrix(gmaps, coordinates, mode="driving", buffer_minutes=5):
    """
    Build the distance matrix between locations using Google Maps API.
    Includes robust handling for missing data and fallback strategies.
    """
    departure_time = (
        int(time.time()) + buffer_minutes * 60 if mode == "transit" else "now"
    )

    n = len(coordinates)
    matrix = np.zeros((n, n))

    for i in range(n):
        for j in range(n):
            if i == j:
                matrix[i][j] = 0  # Zero time for same location
                continue

            try:
                # Fetch travel time for the specified mode
                result = gmaps.distance_matrix(
                    origins=[coordinates[i]],
                    destinations=[coordinates[j]],
                    mode=mode,
                    departure_time=departure_time,
                )
                element = result["rows"][0]["elements"][0]

                if element["status"] == "OK":
                    matrix[i][j] = element["duration"][
                        "value"
                    ]  # Travel time in seconds
                else:
                    raise ValueError(f"Route not available: {element['status']}")
            except Exception:
                # Fallback to walking mode if transit or driving fails
                try:
                    result = gmaps.distance_matrix(
                        origins=[coordinates[i]],
                        destinations=[coordinates[j]],
                        mode="walking",
                    )
                    element = result["rows"][0]["elements"][0]
                    if element["status"] == "OK":
                        matrix[i][j] = element["duration"]["value"]
                    else:
                        matrix[i][j] = np.inf  # Mark as unreachable
                except Exception:
                    matrix[i][j] = np.inf  # Mark as unreachable if walking also fails
                    print(f"Failed to fetch distance between {i} and {j} for␣
 ↪all modes")

    # Replace np.inf with a large penalty value to allow the solver to work
    penalty = 1e6  # High penalty for unreachable routes
    matrix[np.isinf(matrix)] = penalty

    return matrix


def solve_tsp(distance_matrix):
    """
    Solve the Traveling Salesman Problem (TSP) using CVXPY with MTZ constraints.

    Args:
        distance_matrix (np.ndarray): Matrix of travel times.

    Returns:
        tuple: (Tour matrix indicating optimal paths, optimal cost).
    """
    n = distance_matrix.shape[0]
    x = cp.Variable((n, n), boolean=True)
    u = cp.Variable(n)

    # Objective: Minimize total travel time
    objective = cp.Minimize(cp.sum(cp.multiply(distance_matrix, x)))

    # Constraints
    constraints = []

    # Each location must be visited exactly once
    constraints += [cp.sum(x, axis=0) == 1]
    constraints += [cp.sum(x, axis=1) == 1]

    # Avoid self-loops
    for i in range(n):
        constraints.append(x[i, i] == 0)

    # Subtour elimination (MTZ constraints)
    for i in range(1, n):
        for j in range(1, n):
            if i != j:
                constraints.append(u[i] - u[j] + n * x[i, j] <= n - 1)

    for i in range(1, n):
        constraints.append(u[i] >= 2)
        constraints.append(u[i] <= n)
    # Solve the problem
    problem = cp.Problem(objective, constraints)
    problem.solve(solver=cp.GLPK_MI)

    # Extract the solution
    tour_matrix = np.round(x.value)
    return tour_matrix, problem.value


def extract_tour(tour_matrix):
    """
    Extract the optimal tour from the tour matrix.

    Args:
        tour_matrix (np.ndarray): Matrix indicating the optimal paths.

    Returns:
        list: Sequence of indices representing the optimal route.
    """
    n = len(tour_matrix)
    route = []
    current = 0
    while len(route) < n:
        route.append(current)
        next_step = np.argmax(tour_matrix[current])
        tour_matrix[current] = 0  # Mark as visited
        current = next_step
    route.append(route[0])  # Return to start
    return route


def pretty_print_route(optimal_route, total_time):
    """
    Pretty print the optimal route in a user-friendly format, including total␣
 ↪time.

    Args:
        optimal_route (list): List of location names representing the optimal␣
 ↪route.
        total_time (float): Total travel time in seconds.

    Returns:
        str: Formatted string describing the optimal route with total travel␣
 ↪time.
    """
    # Convert total time from seconds to hours, minutes, and seconds
```

```python
    hours, remainder = divmod(int(total_time), 3600)
    minutes, seconds = divmod(remainder, 60)

    # Format route description
    route_description = " -> ".join(optimal_route)

    # Return formatted output
    return f"Total time: {hours} hrs, {minutes} mins, {seconds} seconds\nThe optimal route is: {route_description}"


def get_directions(gmaps, route, locations, mode="driving"):
    """
    Fetch turn-by-turn directions for the optimal route.

    Args:
        gmaps (googlemaps.Client): Google Maps API client.
        route (list): Optimal route indices.
        locations (dict): Dictionary of place names and coordinates.
        mode (str): Transportation mode ("driving", "walking", or "transit").

    Returns:
        list: List of Google Maps directions responses for each leg.
    """
    directions = []
    for i in range(len(route) - 1):
        # Get origin and destination for the segment
        origin = locations[route[i]]
        destination = locations[route[i + 1]]

        # Format as strings for API
        origin_str = f"{origin[0]},{origin[1]}"
        destination_str = f"{destination[0]},{destination[1]}"

        # Fetch directions for the segment
        response = gmaps.directions(
            origin=origin_str, destination=destination_str, mode=mode
        )
        directions.append(response)
    return directions


def parse_directions(directions, route):
    """
    Parse directions from Google Maps API response into a user-friendly format.

    Args:
        directions (list): List of Google Maps API directions responses for␣
↪each leg.
        route (list): List of location names in the optimal route.

    Returns:
        str: User-friendly directions string.
    """
    parsed_directions = []

    for i, segment in enumerate(directions):
        if not segment:
            parsed_directions.append(
                f"No directions available for segment {i + 1}: {route[i]} ->␣
↪{route[i + 1]}"
            )
            continue

        leg = segment[0]["legs"][0]
        start = route[i]
        end = route[i + 1]
        distance = leg["distance"]["text"]
        duration = leg["duration"]["text"]

        # Add header for the segment
        parsed_directions.append(f"\n--- From: {start} To: {end} ---\n")
        parsed_directions.append(f"Distance: {distance}, Duration:␣
↪{duration}\n")

        # Add step-by-step instructions
        for step in leg["steps"]:
            instruction = step["html_instructions"]
            step_distance = step["distance"]["text"]
            step_duration = step["duration"]["text"]

            # Clean HTML tags from instructions
            clean_instruction = (
                instruction.replace("<b>", "")
                .replace("</b>", "")
                .replace('<div style="font-size:0.9em">', " ")
                .replace("</div>", "")
            )

            # Include transit-specific details if applicable
            if "transit_details" in step:
                transit = step["transit_details"]
                line_name = transit["line"]["name"]
                vehicle_type = transit["line"]["vehicle"]["type"]
                departure_stop = transit["departure_stop"]["name"]
                arrival_stop = transit["arrival_stop"]["name"]

                parsed_directions.append(
                    f"- Take {vehicle_type} ({line_name}) from {departure_stop}␣
 ↪to {arrival_stop} ({step_distance}, {step_duration})"
                )
            else:
                parsed_directions.append(
                    f"- {clean_instruction} ({step_distance}, {step_duration})"
                )

    return "\n".join(parsed_directions)


starting_location_name = "Minerva Residence"
starting_location_coords = (37.792033, -122.408465)


destinations = {
    "Chinatown San Francisco": (37.792597, -122.406063),
    "California Academy of Sciences": (37.76986, -122.46609),
    "Pier 39": (37.80867, -122.40982),
    "Painted Ladies": (37.77625, -122.43275),
    "Exploratorium": (37.80166, -122.39734),
    "Lombard Street": (37.80201, -122.41955),
    "Palace of Fine Arts": (37.80293, -122.44842),
    "San Francisco City Hall": (37.77927, -122.41924),
}

# Com
```

8.1 San Francisco locations

[4]:

```
starting_location_name = "Minerva Residence"
starting_location_coords = (37.792033, -122.408465)


destinations = {
    "Chinatown San Francisco": (37.792597, -122.406063),
    "California Academy of Sciences": (37.76986, -122.46609),
    "Pier 39": (37.80867, -122.40982),
    "Painted Ladies": (37.77625, -122.43275),
    "Exploratorium": (37.80166, -122.39734),
    "Lombard Street": (37.80201, -122.41955),
    "Palace of Fine Arts": (37.80293, -122.44842),
    "San Francisco City Hall": (37.77927, -122.41924),
}

# Combine starting, ending, and destinations
all_locations = {starting_location_name: starting_location_coords,␣
 ↪**destinations}

place_names = list(all_locations.keys())
coordinates = list(all_locations.values())
```

[5]:

```
# Transit times For report
mode_1 = "transit"

print(f"--- Testing Mode: {mode_1.upper()} ---")

print("Current time", datetime.now().isoformat())
# Calculate distance matrix
distance_matrix_1 = get_distance_matrix(gmaps, coordinates, mode=mode_1)
```

```
--- Testing Mode: TRANSIT ---
Current time 2024-12-15T05:48:07.789604
```

[20]:

```
from IPython.display import Markdown, display


def print_distance_matrix_markdown(location_names, distance_matrix):
    print("Cost matrix (rounded down in minutes), and 'M' for big M (inf):")
    header = "| " + " | ".join([""] + location_names) + " |"
    separator = "|---" * (len(location_names) + 1) + "|"
    rows = []
    for i, row in enumerate(distance_matrix):
        # if the distance is np.inf or 0, replace it with "M" (for big M)
        row = ["M" if x == np.inf or x == 0.0 else x for x in row]

        # convert to minutes for better readability
        row = [int(x / 60) if x != "M" else "M" for x in row]
        row_str = "| " + location_names[i] + " | " + " | ".join(map(str, row))␣
 ↪+ " |"
        rows.append(row_str)
    table = "\n".join([header, separator] + rows)
    display(Markdown(table))


print_distance_matrix_markdown(place_names, distance_matrix_1)
```

```
Cost matrix (rounded down in minutes), and 'M' for big M (inf):
```

	Minerva Residence	Chinatown San Francisco	California Academy of Sciences	Pier 39	Painted Ladies	Exploratorium	Lombard Street	Palace of Fine Arts	San Francisco City Hall
Minerva Residence	M	4	47	24	33	22	14	40	27
Chinatown San Francisco	5	M	51	23	33	19	19	40	24
California Academy of Sciences	49	47	M	45	25	51	44	28	30
Pier 39	21	20	58	M	50	14	20	27	41
Painted Ladies	34	34	25	45	M	41	50	41	14
Exploratorium	32	29	62	10	48	M	27	47	33
Lombard Street	12	16	53	15	40	28	M	23	24
Palace of Fine Arts	40	42	41	27	39	47	28	M	25
San Francisco City Hall	24	23	33	36	15	33	25	26	M

[10]:

```python
# Driving mode
mode = "driving"

print(f"--- Testing Mode: {mode.upper()} ---")

print("Current time", datetime.now().isoformat())
# Calculate distance matrix
distance_matrix = get_distance_matrix(gmaps, coordinates, mode=mode)

# Solve TSP and extract route
tour_matrix, optimal_cost = solve_tsp(distance_matrix)
optimal_route_indices = extract_tour(tour_matrix)
optimal_route = [place_names[i] for i in optimal_route_indices]

# Print the pretty route
```

```python
print(pretty_print_route(optimal_route, optimal_cost))

# Fetch and display directions
directions = get_directions(gmaps, optimal_route, all_locations, mode=mode)
parsed_directions = parse_directions(directions, optimal_route)
print("Directions:\n", parsed_directions)
```

```
--- Testing Mode: DRIVING ---
Current time 2024-12-15T05:56:31.177570
Total time: 1 hrs, 15 mins, 12 seconds
The optimal route is: Minerva Residence -> Chinatown San Francisco -> San
Francisco City Hall -> Painted Ladies -> California Academy of Sciences ->
Palace of Fine Arts -> Lombard Street -> Pier 39 -> Exploratorium -> Minerva
Residence
Directions:

--- From: Minerva Residence To: Chinatown San Francisco ---

Distance: 0.3 mi, Duration: 3 mins

- Head north on Joice St toward California St (33 ft, 1 min)
- Turn right at the 1st cross street onto California St (0.1 mi, 1 min)
- Turn right onto Quincy St (354 ft, 1 min)
- Turn right onto Pine St (89 ft, 1 min)
- Turn right at the 1st cross street onto Grant Ave Parts of this road may be
closed at certain times or days Destination will be on the left (404 ft, 1 min)

--- From: Chinatown San Francisco To: San Francisco City Hall ---

Distance: 1.8 mi, Duration: 12 mins

- Head north on Grant Ave toward Sacramento St May be closed at certain times or
days (292 ft, 1 min)
- Turn left at the 1st cross street onto Sacramento St (0.2 mi, 2 mins)
- Turn left onto Powell St (0.1 mi, 1 min)
- Turn right at the 2nd cross street onto Pine St (0.5 mi, 2 mins)
- Turn left onto Hyde St Parts of this road may be closed at certain times or
days (0.7 mi, 4 mins)
- Turn right onto McAllister St (0.2 mi, 1 min)
- Turn left onto Polk St (0.1 mi, 1 min)
- Turn right onto Grove St (213 ft, 1 min)
- Turn right Destination will be on the right (161 ft, 1 min)

--- From: San Francisco City Hall To: Painted Ladies ---

Distance: 0.8 mi, Duration: 5 mins

- Head west toward Grove St (151 ft, 1 min)
- Turn right onto Grove St (0.8 mi, 5 mins)
- Turn left onto Steiner St Destination will be on the left (128 ft, 1 min)

--- From: Painted Ladies To: California Academy of Sciences ---

Distance: 2.6 mi, Duration: 10 mins

- Head south on Steiner St toward Hayes St (0.1 mi, 1 min)
- Turn right onto Fell St (1.1 mi, 4 mins)
- Slight left to stay on Fell St (0.1 mi, 1 min)
- Continue onto John F Kennedy Dr (335 ft, 1 min)
- Keep left to continue on Kezar Dr (0.4 mi, 1 min)
- Turn right onto Martin Luther King Jr Dr (0.5 mi, 2 mins)
- Turn right to stay on Martin Luther King Jr Dr (0.2 mi, 1 min)
- Turn right onto Music Concourse Dr (0.1 mi, 1 min)

--- From: California Academy of Sciences To: Palace of Fine Arts ---

Distance: 5.0 mi, Duration: 14 mins

- Head northeast on Music Concourse Dr (371 ft, 1 min)
- Music Concourse Dr turns slightly left and becomes Bowl Dr (335 ft, 1 min)
- Turn left onto Hagiwara Tea Garden Dr (0.2 mi, 2 mins)
- Turn right onto Martin Luther King Jr Dr (0.5 mi, 2 mins)
- Turn right onto CA-1 N (2.8 mi, 6 mins)
- Slight right onto the US-101 S ramp to Marina Blvd (0.3 mi, 1 min)
- Merge onto US-101 (0.6 mi, 1 min)
- Take exit 437 toward Presidio/<wbr/>Marina Blvd (0.1 mi, 1 min)
- Take the Girard Rd ramp (121 ft, 1 min)
- Turn left onto Girard Rd (0.2 mi, 1 min)
- Turn right (0.1 mi, 1 min)

--- From: Palace of Fine Arts To: Lombard Street ---

Distance: 1.8 mi, Duration: 10 mins

- Head south toward Lyon St (0.1 mi, 1 min)
- Turn right onto Lyon St (213 ft, 1 min)
- Turn left onto Richardson Ave (0.2 mi, 1 min)
- Continue onto Lombard St (1.1 mi, 5 mins)
- Continue straight past Comfort Inn By the Bay to stay on Lombard St (0.3 mi, 2
mins)

--- From: Lombard Street To: Pier 39 ---

Distance: 1.0 mi, Duration: 8 mins

- Head northeast on Lombard St toward Montclair Terrace (203 ft, 1 min)
- Turn right to stay on Lombard St (384 ft, 1 min)
- Turn left onto Leavenworth St (0.3 mi, 2 mins)
- Turn right onto North Point St (0.5 mi, 3 mins)
- Turn left onto Grant Ave (335 ft, 1 min)
- Turn left onto The Embarcadero N (26 ft, 1 min)
- Turn right Destination will be on the right (236 ft, 1 min)

--- From: Pier 39 To: Exploratorium ---

Distance: 0.9 mi, Duration: 7 mins

- Head west toward Pier 39/<wbr/>Pier 39 Concourse (295 ft, 1 min)
- Turn left onto The Embarcadero N Pass by Hillstone (on the right in 0.4 mi)
(0.9 mi, 6 mins)

--- From: Exploratorium To: Minerva Residence ---

Distance: 1.1 mi, Duration: 8 mins

- Head southwest on Green St toward The Embarcadero N (0.1 mi, 1 min)
- Turn left onto Battery St (0.5 mi, 3 mins)
- Turn right onto California St Pass by Wells Fargo Bank (on the right) (0.5 mi,
3 mins)
- Turn left onto Joice St (33 ft, 1 min)
```

[13]:

```python
# Transit mode
mode = "walking"

print(f"--- Testing Mode: {mode.upper()} ---")

print("Current time", datetime.now().isoformat())
# Calculate distance matrix
distance_matrix = get_distance_matrix(gmaps, coordinates, mode=mode)

# Solve TSP and extract route
tour_matrix, optimal_cost = solve_tsp(distance_matrix)
optimal_route_indices = extract_tour(tour_matrix)
optimal_route = [place_names[i] for i in optimal_route_indices]

# Print the pretty route
print(pretty_print_route(optimal_route, optimal_cost))

# Fetch and display directions
directions = get_directions(gmaps, optimal_route, all_locations, mode=mode)
parsed_directions = parse_directions(directions, optimal_route)
print("Directions:\n", parsed_directions)
```

```
--- Testing Mode: WALKING ---
```

```
Current time 2024-12-14T13:13:37.088254
Total time: 4 hrs, 47 mins, 58 seconds
The optimal route is: Minerva Residence -> Chinatown San Francisco ->
Exploratorium -> Pier 39 -> Lombard Street -> Palace of Fine Arts -> California
Academy of Sciences -> Painted Ladies -> San Francisco City Hall -> Minerva
Residence
Directions:

--- From: Minerva Residence To: Chinatown San Francisco ---

Distance: 0.2 mi, Duration: 4 mins

- Head east on California St toward Joice St (0.1 mi, 4 mins)
- Turn left onto Grant Ave Destination will be on the left (52 ft, 1 min)

--- From: Chinatown San Francisco To: Exploratorium ---

Distance: 1.1 mi, Duration: 23 mins

- Head north on Grant Ave toward Sacramento St (0.3 mi, 6 mins)
- Turn right onto Pacific Ave (0.3 mi, 7 mins)
- Turn left onto Battery St (0.2 mi, 4 mins)
- Turn right onto Green St (0.1 mi, 3 mins)
- Turn left onto The Embarcadero N (30 ft, 1 min)
- Turn right onto Pier 15 Destination will be on the right (0.1 mi, 2 mins)

--- From: Exploratorium To: Pier 39 ---

Distance: 0.9 mi, Duration: 20 mins

- Head southwest on Pier 15 toward The Embarcadero N (0.1 mi, 2 mins)
- Turn right onto The Embarcadero N (0.7 mi, 16 mins)
- Turn right toward Pier 39 (46 ft, 1 min)
- Turn left toward Pier 39 (358 ft, 1 min)
- Turn right onto Pier 39 (62 ft, 1 min)
- Turn right (30 ft, 1 min)

--- From: Pier 39 To: Lombard Street ---

Distance: 1.0 mi, Duration: 25 mins

- Head west toward Pier 39 (30 ft, 1 min)
- Turn left onto Pier 39 (102 ft, 1 min)
- Turn right toward Beach St (210 ft, 1 min)
- Slight left toward Beach St (167 ft, 1 min)
- Turn right onto Beach St (0.5 mi, 10 mins)
- Turn left onto Leavenworth St (0.3 mi, 10 mins)
- Turn right onto Hyde St Take the stairs Destination will be on the right (0.1
mi, 3 mins)

--- From: Lombard Street To: Palace of Fine Arts ---

Distance: 1.8 mi, Duration: 40 mins

- Head west on Lombard St toward Larkin St Pass by Chase Bank (on the left in
0.3 mi) (1.4 mi, 32 mins)
- Continue onto Richardson Ave (299 ft, 1 min)
- Turn right toward Richardson Ave (43 ft, 1 min)
- Take the crosswalk (82 ft, 1 min)
- Turn right onto Richardson Ave (0.1 mi, 3 mins)
- Turn right onto Lyon St Take the stairs (0.1 mi, 2 mins)
- Turn left (262 ft, 1 min)
- Turn left (102 ft, 1 min)

--- From: Palace of Fine Arts To: California Academy of Sciences ---

Distance: 3.1 mi, Duration: 1 hour 15 mins

- Head south (102 ft, 1 min)
- Turn right toward Lyon St (262 ft, 1 min)
- Turn right onto Lyon St Take the stairs (364 ft, 2 mins)
- Slight right to stay on Lyon St (167 ft, 1 min)
- Continue onto Gorgas Ave (0.1 mi, 3 mins)
- Turn left onto O'Reilly Ave (0.1 mi, 3 mins)
- Turn left toward Letterman Dr (13 ft, 1 min)
- Turn right toward Letterman Dr (381 ft, 2 mins)
- Turn left toward Letterman Dr (36 ft, 1 min)
- Turn right onto Letterman Dr (226 ft, 1 min)
- Continue onto Presidio Blvd (308 ft, 1 min)
- Turn left onto MacArthur Ave (0.2 mi, 5 mins)
- Turn left to stay on MacArthur Ave (0.2 mi, 4 mins)
- Slight left to stay on MacArthur Ave (66 ft, 1 min)
- Slight left to stay on MacArthur Ave (138 ft, 1 min)
- Turn right Take the stairs (495 ft, 3 mins)
- Turn left toward Ecology Trail (0.1 mi, 3 mins)
- Continue onto Ecology Trail (0.1 mi, 5 mins)
- Turn left toward Arguello Blvd (102 ft, 1 min)
- Turn right toward Arguello Blvd (39 ft, 1 min)
- Turn left onto Arguello Blvd Pass by Office Depot (on the right in 0.6 mi)
(1.2 mi, 28 mins)
- Continue onto Conservatory Dr W (0.1 mi, 2 mins)
- Turn right (23 ft, 1 min)
- Turn left (420 ft, 2 mins)
- Turn right (36 ft, 1 min)
- Turn left (348 ft, 2 mins)
- Turn left (0.1 mi, 3 mins)
- Turn right (69 ft, 1 min)
- Turn left (33 ft, 1 min)
- Turn left (427 ft, 2 mins)
- Turn left (10 ft, 1 min)

--- From: California Academy of Sciences To: Painted Ladies ---

Distance: 2.0 mi, Duration: 45 mins

- Head northwest (10 ft, 1 min)
- Turn right (427 ft, 2 mins)
- Turn right (33 ft, 1 min)
- Turn right (69 ft, 1 min)
- Turn left (0.1 mi, 3 mins)
- Turn right toward 3rd Ave (348 ft, 2 mins)
- Turn right toward 3rd Ave (161 ft, 1 min)
- Turn left onto 3rd Ave (348 ft, 1 min)
- Turn right toward Hayes St (180 ft, 1 min)
- Continue onto Hayes St (138 ft, 1 min)
- Turn left (0.2 mi, 5 mins)
- Continue onto Hayes St Pass by Popeyes Louisiana Kitchen (on the right in 1
mi) (1.3 mi, 29 mins)
- Turn left onto Steiner St Destination will be on the right (213 ft, 1 min)

--- From: Painted Ladies To: San Francisco City Hall ---

Distance: 0.9 mi, Duration: 21 mins

- Head north on Steiner St toward Grove St (131 ft, 1 min)
- Turn right onto Grove St (0.8 mi, 18 mins)
- Turn left onto Polk St (328 ft, 1 min)
- Turn left Take the stairs (72 ft, 1 min)

--- From: San Francisco City Hall To: Minerva Residence ---

Distance: 1.4 mi, Duration: 34 mins

- Head north toward Polk St (210 ft, 1 min)
- Turn left onto Polk St (121 ft, 1 min)
- Turn right toward McAllister St (52 ft, 1 min)
- Turn left toward McAllister St (49 ft, 1 min)
- Turn right onto McAllister St (0.4 mi, 8 mins)
- Slight left onto Market St (0.3 mi, 8 mins)
- Turn left onto Powell St (72 ft, 1 min)
- Turn right to stay on Powell St Pass by Starbucks (on the right in 0.3 mi)
(0.5 mi, 13 mins)
- Turn right onto Pine St (236 ft, 1 min)
- Turn left onto Joice St Take the stairs Destination will be on the left (312
ft, 2 mins)
```

[14]:

```python
# Transit mode with walking fallback
mode = "transit"
print(f"--- Testing Mode: {mode.upper()} ---")

print("Current time", datetime.now().isoformat())
# Calculate distance matrix
distance_matrix = get_distance_matrix(gmaps, coordinates, mode=mode,␣
 ↪buffer_minutes=5)

# Solve TSP and extract route
tour_matrix, optimal_cost = solve_tsp(distance_matrix)
optimal_route_indices = extract_tour(tour_matrix)
optimal_route = [place_names[i] for i in optimal_route_indices]

print()

# Print the pretty route
print(pretty_print_route(optimal_route, optimal_cost))

# Fetch and display directions
directions = get_directions(gmaps, optimal_route, all_locations, mode=mode)
parsed_directions = parse_directions(directions, optimal_route)
print("Directions:\n", parsed_directions)
```

```
--- Testing Mode: TRANSIT ---
Current time 2024-12-14T13:13:44.598303

Total time: 2 hrs, 44 mins, 29 seconds
The optimal route is: Minerva Residence -> Chinatown San Francisco ->
Exploratorium -> Pier 39 -> Lombard Street -> Palace of Fine Arts -> California
Academy of Sciences -> Painted Ladies -> San Francisco City Hall -> Minerva
Residence
Directions:

--- From: Minerva Residence To: Chinatown San Francisco ---

Distance: 0.2 mi, Duration: 4 mins

- Walk to 607 Grant Ave, San Francisco, CA 94108, USA (0.2 mi, 4 mins)

--- From: Chinatown San Francisco To: Exploratorium ---

Distance: 1.2 mi, Duration: 20 mins

- Walk to Clay St & Grant Ave (0.1 mi, 3 mins)
- Take BUS (California) from Clay St & Grant Ave to Clay St & Drumm St (0.5 mi,
4 mins)
- Walk to Pier 15 Embarcadero at, Green St, San Francisco, CA 94111, USA (0.5
mi, 12 mins)

--- From: Exploratorium To: Pier 39 ---

Distance: 0.9 mi, Duration: 11 mins

- Walk to The Embarcadero & Green St (0.1 mi, 2 mins)
- Take TRAM (Market & Wharves) from The Embarcadero & Green St to The
Embarcadero & Stockton St (0.8 mi, 7 mins)
- Walk to 256 Pier 39, San Francisco, CA 94133, USA (128 ft, 1 min)

--- From: Pier 39 To: Lombard Street ---

Distance: 1.0 mi, Duration: 16 mins

- Walk to Powell St & Beach St (0.2 mi, 4 mins)
- Take BUS (19th Avenue) from Powell St & Beach St to North Point St & Hyde St
(0.5 mi, 5 mins)
- Walk to Hyde St & North Point St (118 ft, 1 min)
- Take CABLE_CAR (Powell-Hyde Cable Car) from Hyde St & North Point St to Hyde
St & Lombard St (0.3 mi, 2 mins)

--- From: Lombard Street To: Palace of Fine Arts ---

Distance: 1.9 mi, Duration: 22 mins

- Walk to Van Ness Ave & Chestnut St (0.4 mi, 9 mins)
- Take BUS (19th Avenue) from Van Ness Ave & Chestnut St to Richardson Ave &
Francisco St (1.3 mi, 8 mins)
- Walk to 3601 Lyon St, San Francisco, CA 94123, USA (0.2 mi, 5 mins)

--- From: Palace of Fine Arts To: California Academy of Sciences ---

Distance: 5.2 mi, Duration: 32 mins

- Walk to Richardson Ave & Francisco St (0.2 mi, 6 mins)
- Take BUS (19th Avenue) from Richardson Ave & Francisco St to Park Presidio
Blvd & Fulton St (4.5 mi, 21 mins)
- Walk to 55 Music Concourse Dr, San Francisco, CA 94118, USA (0.5 mi, 12 mins)

--- From: California Academy of Sciences To: Painted Ladies ---

Distance: 2.3 mi, Duration: 20 mins

- Walk to Fulton St & 8th Ave (0.3 mi, 7 mins)
- Take BUS (Hayes) from Fulton St & 8th Ave to Hayes St & Steiner St (1.9 mi, 12
mins)
- Walk to 716 Steiner St, San Francisco, CA 94117, USA (272 ft, 1 min)

--- From: Painted Ladies To: San Francisco City Hall ---

Distance: 1.0 mi, Duration: 11 mins

- Walk to Hayes St & Steiner St (272 ft, 1 min)
- Take BUS (Hayes) from Hayes St & Steiner St to Grove St & Van Ness Ave (0.8
mi, 7 mins)
- Walk to 1 Dr Carlton B Goodlett Pl, San Francisco, CA 94102, USA (0.1 mi, 3
mins)

--- From: San Francisco City Hall To: Minerva Residence ---

Distance: 1.6 mi, Duration: 23 mins

- Walk to Civic Center Station (0.3 mi, 7 mins)
- Take TRAM (Ocean View) from Civic Center Station to Powell Station (0.6 mi, 1
min)
- Walk to Powell St & Market St (430 ft, 2 mins)
- Take CABLE_CAR (Powell-Mason Cable Car) from Powell St & Market St to Powell
St & California St (0.5 mi, 7 mins)
- Walk to 851 California St, San Francisco, CA 94108, USA (266 ft, 1 min)
```

8.2 Seoul locations

[20]:

```python
PREDEFINED_DESTINATIONS_Seoul = {
    # "ICN": (37.458896, 126.441946),
    "DDP": (37.567123, 127.010004),
    "COEX": (37.511768, 127.059156),
    "Namsan Mountain Tower": (37.551225, 126.988188),
    "Seoul Station": (37.554859, 126.970783),
    "Jamsil Lotte Tower": (37.512538, 127.102310),
}  #
```

[21]:

```python
# Combine starting, ending, and destinations
all_locations = {
    starting_location_name: starting_location_coords,
    **PREDEFINED_DESTINATIONS_Seoul,
}

starting_location_name = "Minerva Residence (Seoul) - Mangroove Sinseoul"
starting_location_coords = (37.576113730905135, 127.02608903093144)

place_names = list(all_locations.keys())
coordinates = list(all_locations.values())
```

[22]:

```python
mode = "transit"

print(f"--- Testing Mode: {mode.upper()} ---")

print("Current time", datetime.now().isoformat())
# Calculate distance matrix
distance_matrix = get_distance_matrix(gmaps, coordinates, mode=mode)

# Solve TSP and extract route
tour_matrix, optimal_cost = solve_tsp(distance_matrix)
optimal_route_indices = extract_tour(tour_matrix)
optimal_route = [place_names[i] for i in optimal_route_indices]

# Print the pretty route
print(pretty_print_route(optimal_route, optimal_cost))

# Fetch and display directions
directions = get_directions(gmaps, optimal_route, all_locations, mode=mode)
parsed_directions = parse_directions(directions, optimal_route)
print("Directions:\n", parsed_directions)
```

```
--- Testing Mode: TRANSIT ---
Current time 2024-12-14T13:19:53.794466
Total time: 2 hrs, 19 mins, 40 seconds
The optimal route is: Minerva Residence (Seoul) - Mangroove Sinseoul -> Jamsil
Lotte Tower -> COEX -> Seoul Station -> Namsan Mountain Tower -> DDP -> Minerva
Residence (Seoul) - Mangroove Sinseoul
Directions:

--- From: Minerva Residence (Seoul) - Mangroove Sinseoul To: Jamsil Lotte Tower
---

Distance: 11.9 km, Duration: 39 mins

- Walk to Sinseol-dong Station (25 m, 1 min)
- Take BUS (     ) from Sinseol-dong Station to Jamsil Station. Jamsildaegyo
Bridge. South (11.6 km, 33 mins)
- Walk to 300 Olympic-ro, Sincheon-dong, Songpa District, Seoul, South Korea
(0.3 km, 4 mins)

--- From: Jamsil Lotte Tower To: COEX ---

Distance: 4.0 km, Duration: 17 mins

- Walk to Jamsil (0.2 km, 4 mins)
- Take SUBWAY (     2 ) from Jamsil to Samseong (World Trade Center Seoul)
(3.3 km, 6 mins)
- Walk to South Korea, Seoul,    159-7        KR (0.5 km, 8 mins)
```

```
--- From: COEX To: Seoul Station ---

Distance: 15.3 km, Duration: 37 mins

- Walk to Bongeunsa (0.3 km, 5 mins)
- Take SUBWAY (   ) from Bongeunsa to Dongjag (7.6 km, 12 mins)
- Walk to Dongjag (1 m, 1 min)
- Take SUBWAY (   ) from Dongjag to Seoul Station (Subway) (7.2 km, 11 mins)
- Walk to Seoul Station, 405 Hangang-daero, Yongsan District, Seoul, South Korea
(0.2 km, 3 mins)

--- From: Seoul Station To: Namsan Mountain Tower ---

Distance: 3.3 km, Duration: 21 mins

- Walk to Seoul Station 5 Platform (0.2 km, 3 mins)
- Take BUS (    ) from Seoul Station 5 Platform to Huam Mineral Spring (2.7
km, 9 mins)
- Walk to 1-3 yongsan-dong 2(i)-ga, Yongsan District, Seoul, South Korea (0.5
km, 8 mins)

--- From: Namsan Mountain Tower To: DDP ---

Distance: 3.5 km, Duration: 26 mins

- Walk to Myeongdong (1.1 km, 19 mins)
- Take SUBWAY (   ) from Myeongdong to Dongdaemun History & Culture Park (2.1
km, 4 mins)
- Walk to 143 Euljiro 7(chil)-ga, Jung District, Seoul, South Korea (0.2 km, 3
mins)

--- From: DDP To: Minerva Residence (Seoul) - Mangroove Sinseoul ---

Distance: 2.1 km, Duration: 9 mins

- Walk to Dongdaemun History & Culture Park (76 m, 1 min)
- Take BUS (    ) from Dongdaemun History & Culture Park to Sinseol-dong
Station (2.0 km, 6 mins)
- Walk to 22 Wangsan-ro, Dongdaemun District, Seoul, South Korea (71 m, 1 min)
```