

Assignment 2

Turkey Navigation

CMPE 160

Contact: B. İrem Urhan (beyza.urhan@bogazici.edu.tr)

Due:

April 5th, 6 a.m. in the morning

In this assignment, you will develop a Turkey city navigation application using Java. The goal is to find the shortest path between two input cities. You need to implement an efficient path-finding algorithm to find the shortest path, i.e., do not try all permutations, and do not use a random search. You should graphically display the path using the StdDraw graphics library, as shown in the example cases below. Additionally, you are required to output the total distance and the path found for the console output. See the example cases below for the console output format. City information, road information, and Turkey map are provided as input files. In your implementation, a city should be modeled using the City class. The UML class diagram is given in Figure 1. Write your program in such a way that if we use a different city/road input file configuration, it should work correctly, e.g., in our testing we may use different cities and roads.

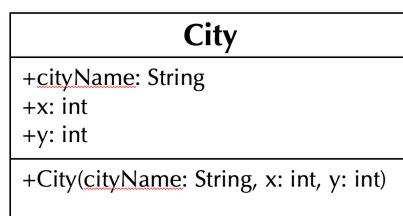
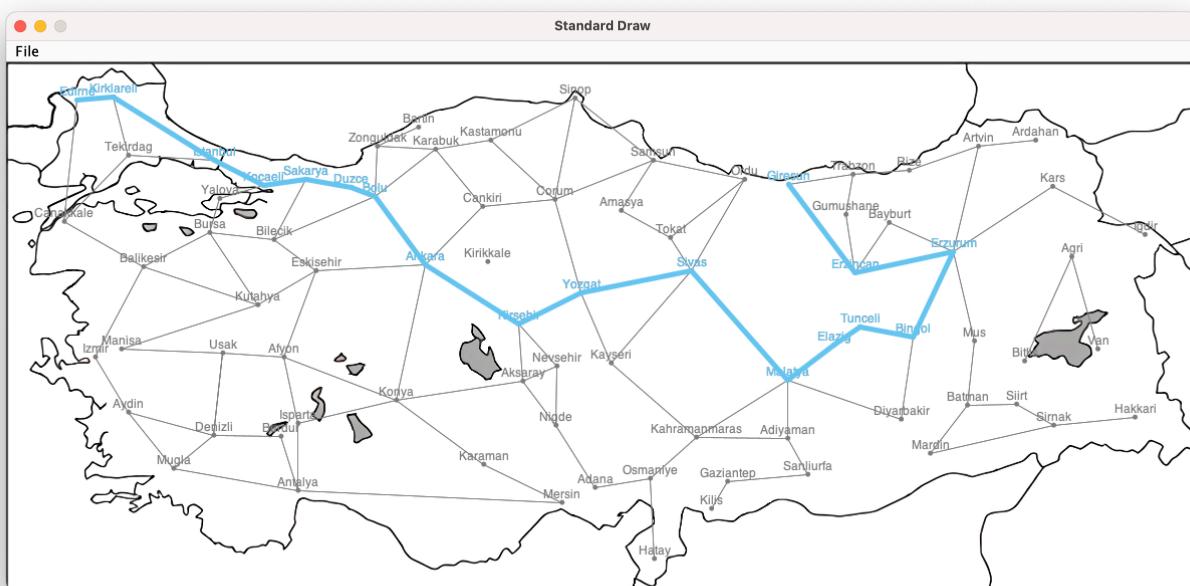


Figure 1. The UML class diagram of the City class.

Case 1: Edirne to Giresun

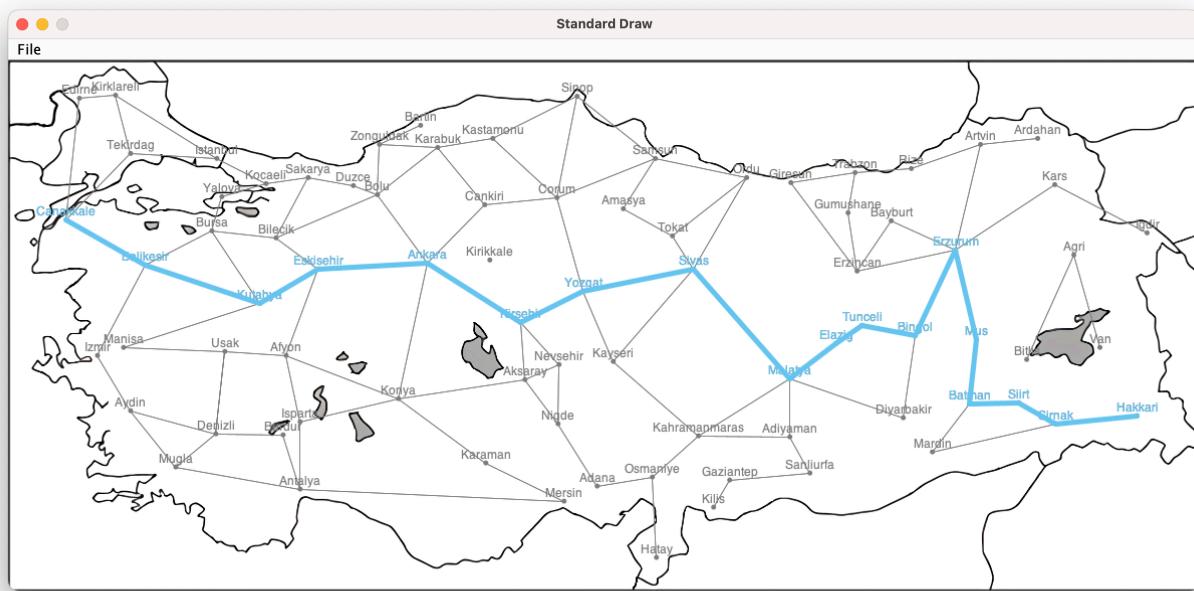


Enter starting city: Edirne

Enter destination city: Giresun

Total Distance: 2585.49. Path: Edirne -> Kırklareli -> İstanbul -> Kocaeli -> Sakarya -> Duzce -> Bolu -> Ankara -> Kirşehir -> Yozgat -> Sivas -> Malatya -> Elazığ -> Tunceli -> Bingöl -> Erzurum -> Erzincan -> Giresun

Case 2: Çanakkale to Hakkari

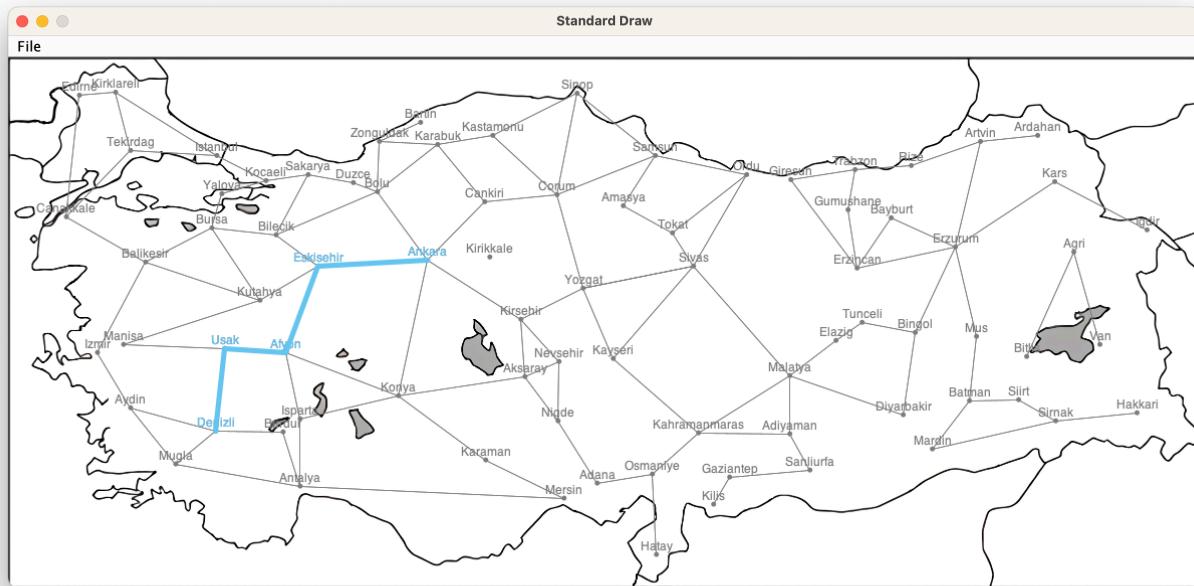


Enter starting city: Çanakkale

Enter destination city: Hakkari

Total Distance: 2780.87. Path: Çanakkale -> Balikesir -> Kutahya -> Eskisehir -> Ankara -> Kirsehir -> Yozgat -> Sivas -> Malatya -> Elazig -> Tunceli -> Bingol -> Erzurum -> Mus -> Batman -> Siirt -> Sirnak -> Hakkari

Case 3: Invalid city names: User should be prompted again to enter a valid city name.



Enter starting city: Anka

City named 'Anka' not found. Please enter a valid city name.

Enter starting city: Ankara

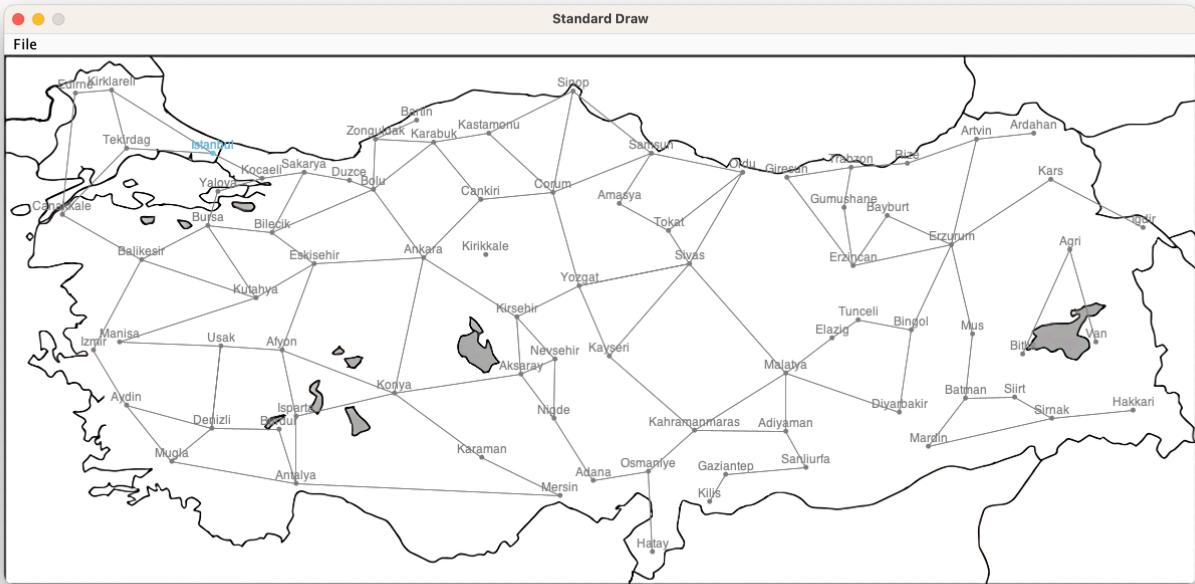
Enter destination city: Deni

City named 'Deni' not found. Please enter a valid city name.

Enter destination city: Denizli

Total Distance: 689.10. Path: Ankara -> Eskisehir -> Afyon -> Usak -> Denizli

Case 4: Path to the same city



Enter starting city: İstanbul

Enter destination city: İstanbul

Total Distance: 0.00. Path: İstanbul

Case 5: Unreachable city pairs: There may be no path from the starting city to the destination city. In this case, no graphical output is produced and console output should be "No path could be found".

Enter starting city: Izmir

Enter destination city: Van

No path could be found.

Report Contents:

In your report,

1. Provide outputs (graphical and console outputs) as shown in this document, for each case shown above.
2. Explain your path-finding algorithm in detail. Additionally, provide pseudocode of the path-finding algorithm. Provide references.

Evaluation Criteria

Correctness of the distance and path outputs.

Effective use of OOP and Java programming best practices.

Application's robustness in terms of error handling and user input validation.

Overall code readability and documentation, including coding practices and comments.

Submission Guide

Submission Files

Submit a single compressed (.zip) file, named as name_surname.zip, to the Moodle. It should contain all source code files (under the \code directory), report (in PDF format, under the \report directory) and all other files if needed (under \misc directory)

File Naming

Name the main code as "NameSurname.java". Name your report as "NameSurname.pdf".

Late Submission Policy

Maximum delay is two days. Late submission will be graded on a scale of 50% of the original grade.

Mandatory Submission

Submission of assignments is mandatory. If you do not submit an assignment, you will fail the course.

Plagiarism

Leads to grade F and YÖK regulations will be applied.