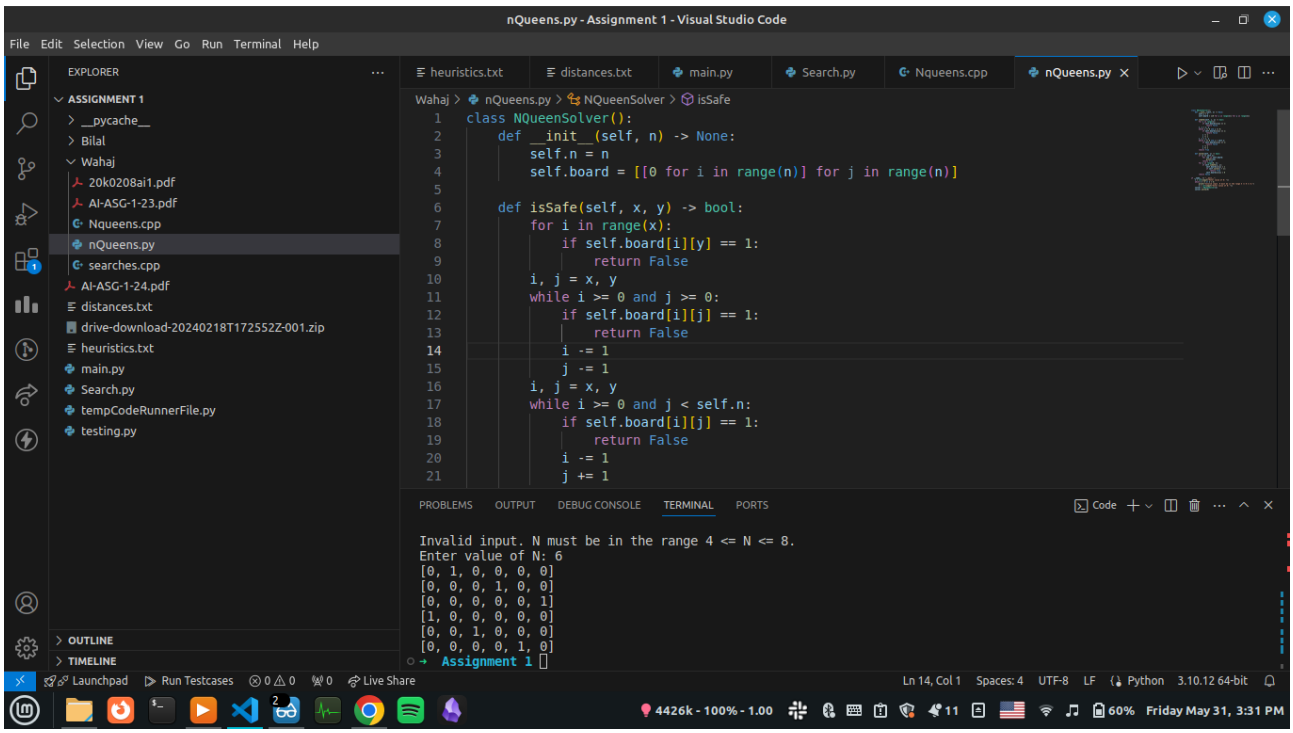
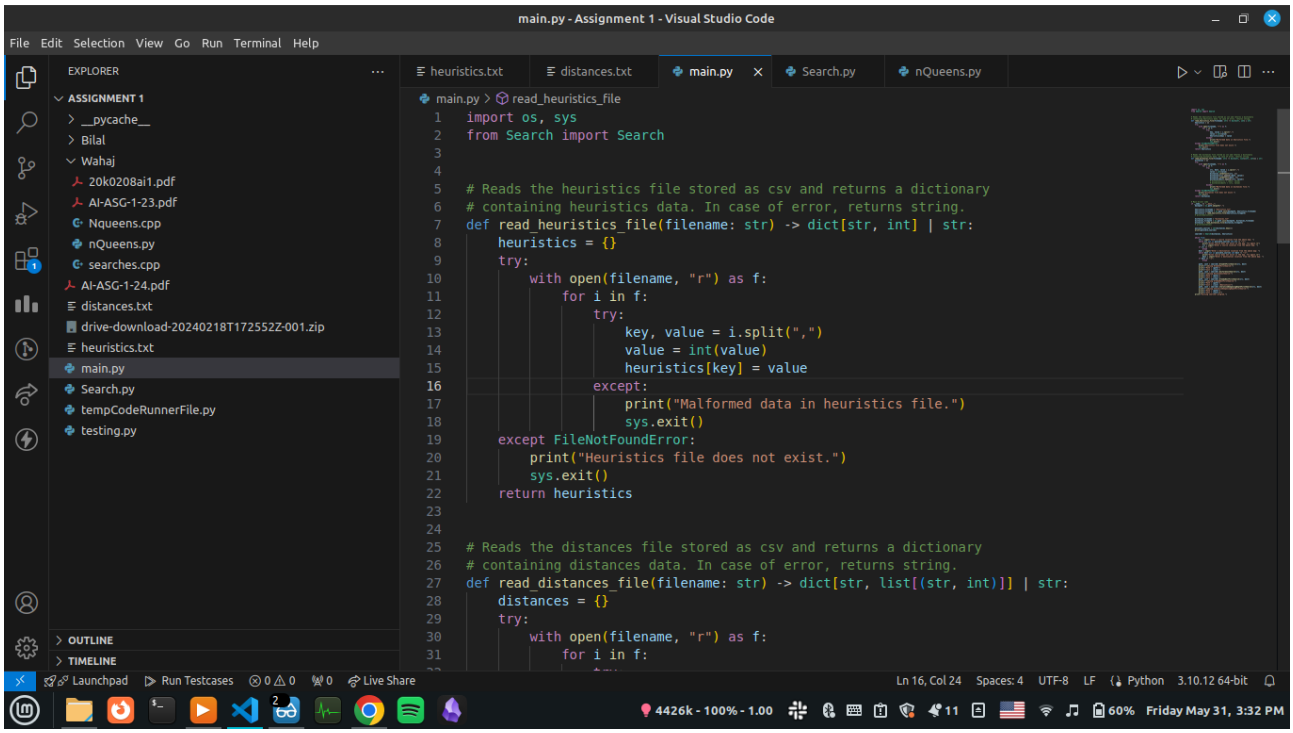
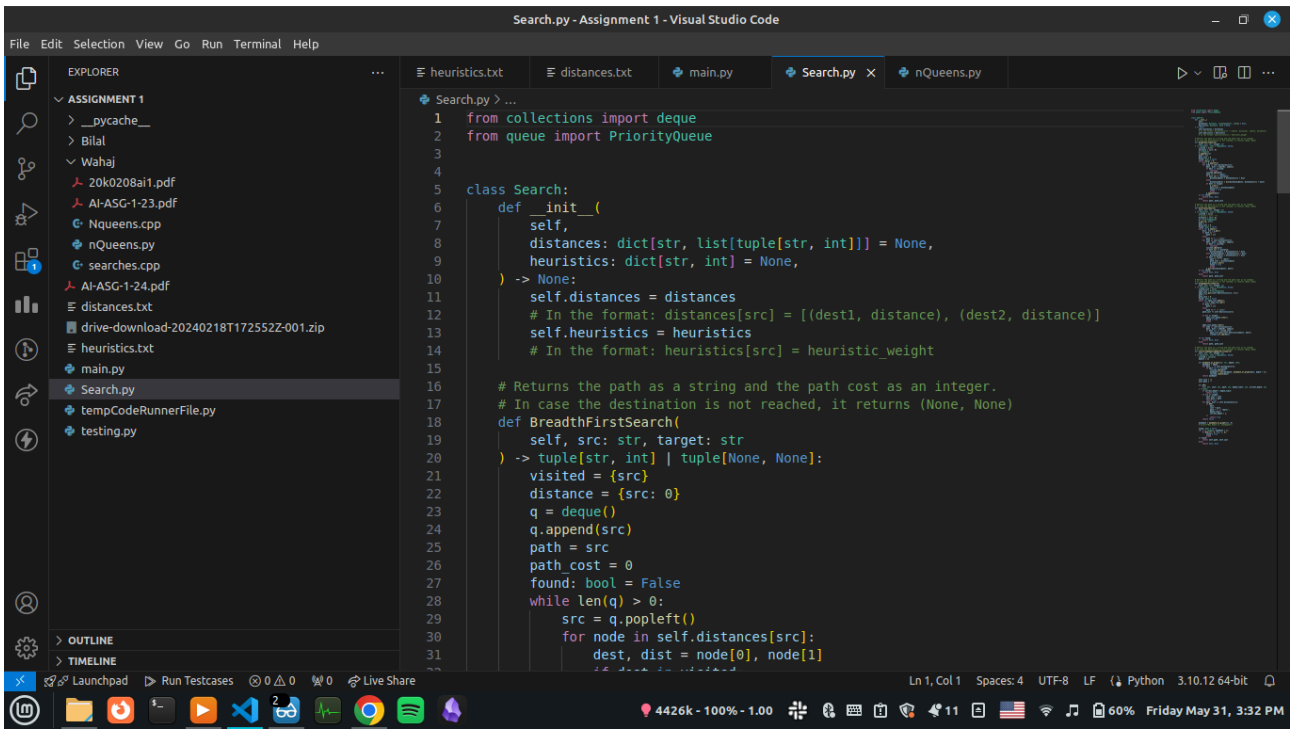
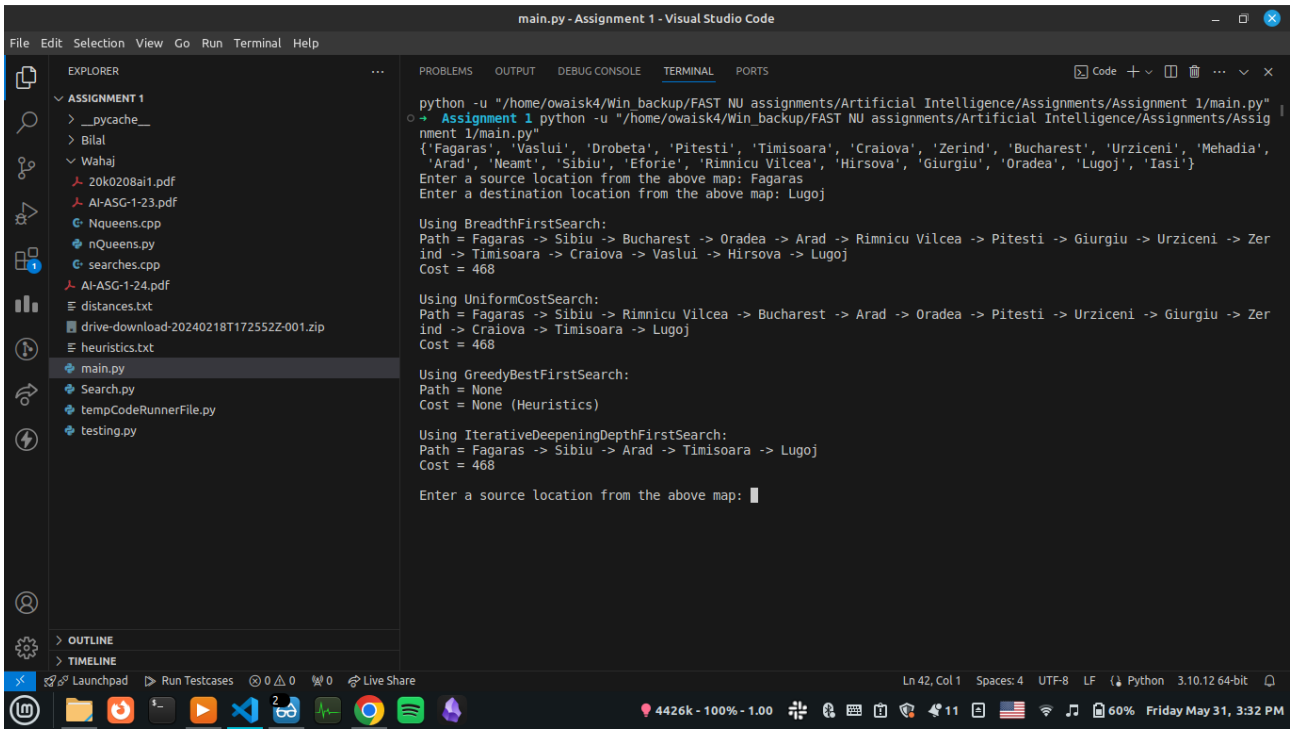

nQueens.py - Assignment 1 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
ASSIGNMENT 1
__pycache__
Bilal
Wahaj
20k0208ai1.pdf
AI-ASG-1-23.pdf
Nqueens.cpp
nQueens.py
searches.cpp
AI-ASG-1-24.pdf
distances.txt
drive-download-20240218T172552Z-001.zip
heuristics.txt
main.py
Search.py
tempCodeRunnerFile.py
testing.py
OUTLINE
TIMELINE
heuristics.txt
distances.txt
main.py
Search.py
Nqueens.cpp
nQueens.py
Wahaj > nQueens.py > NQueenSolver > isSafe
class NQueenSolver():
def __init__(self, n) -> None:
self.n = n
self.board = [[0 for i in range(n)] for j in range(n)]
def isSafe(self, x, y) -> bool:
for i in range(x):
if self.board[i][y] == 1:
return False
i, j = x, y
while i >= 0 and j >= 0:
if self.board[i][j] == 1:
return False
i -= 1
j -= 1
i, j = x, y
while i >= 0 and j < self.n:
if self.board[i][j] == 1:
return False
i -= 1
j += 1
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code
Invalid input. N must be in the range 4 <= N <= 8.
Enter value of N: 6
[0, 1, 0, 0, 0, 0]
[0, 0, 0, 1, 0, 0]
[0, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 0, 0]
[0, 0, 1, 0, 0, 0]
[0, 0, 0, 0, 1, 0]
Assignment 1
Launchpad Run Testcases 0 0 0 Live Share
Ln 14, Col 1 Spaces: 4 UTF-8 LF Python 3.10.12 64-bit
4426k - 100% - 1.00 11 60% Friday May 31, 3:31 PM


main.py - Assignment 1 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
ASSIGNMENT 1
__pycache__
Bilal
Wahaj
20k0208ai1.pdf
AI-ASG-1-23.pdf
Nqueens.cpp
nQueens.py
searches.cpp
AI-ASG-1-24.pdf
distances.txt
drive-download-20240218T172552Z-001.zip
heuristics.txt
main.py
Search.py
tempCodeRunnerFile.py
testing.py
OUTLINE
TIMELINE
heuristics.txt
distances.txt
main.py
Search.py
nQueens.py
main.py > read_heuristics_file
import os, sys
from Search import Search
# Reads the heuristics file stored as csv and returns a dictionary
# containing heuristics data. In case of error, returns string.
def read_heuristics_file(filename: str) -> dict[str, int] | str:
heuristics = {}
try:
with open(filename, "r") as f:
for i in f:
try:
key, value = i.split(",")
value = int(value)
heuristics[key] = value
except:
print("Malformed data in heuristics file.")
sys.exit()
except FileNotFoundError:
print("Heuristics file does not exist.")
sys.exit()
return heuristics
# Reads the distances file stored as csv and returns a dictionary
# containing distances data. In case of error, returns string.
def read_distances_file(filename: str) -> dict[str, list[(str, int)]] | str:
distances = {}
try:
with open(filename, "r") as f:
for i in f:
Launchpad Run Testcases 0 0 0 Live Share
Ln 16, Col 24 Spaces: 4 UTF-8 LF Python 3.10.12 64-bit
4426k - 100% - 1.00 11 60% Friday May 31, 3:32 PM


Search.py - Assignment 1 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
ASSIGNMENT 1
__pycache__
Bilal
Wahaj
20k0208ai1.pdf
AI-ASG-1-23.pdf
Nqueens.cpp
nQueens.py
searches.cpp
AI-ASG-1-24.pdf
distances.txt
drive-download-20240218T172552Z-001.zip
heuristics.txt
main.py
Search.py
tempCodeRunnerFile.py
testing.py
OUTLINE
TIMELINE
heuristics.txt
distances.txt
main.py
Search.py
nQueens.py
Search.py > ...
from collections import deque
from queue import PriorityQueue
class Search:
def __init__(
self,
distances: dict[str, list[tuple[str, int]]] = None,
heuristics: dict[str, int] = None,
) -> None:
self.distances = distances
# In the format: distances[src] = [(dest1, distance), (dest2, distance)]
self.heuristics = heuristics
# In the format: heuristics[src] = heuristic_weight
# Returns the path as a string and the path cost as an integer.
# In case the destination is not reached, it returns (None, None)
def BreadthFirstSearch(
self, src: str, target: str
) -> tuple[str, int] | tuple[None, None]:
visited = {src}
distance = {src: 0}
q = deque()
q.append(src)
path = src
path_cost = 0
found: bool = False
while len(q) > 0:
src = q.popleft()
for node in self.distances[src]:
dest, dist = node[0], node[1]
Launchpad
Run Testcases
Live Share
Ln 1, Col 1
Spaces: 4
UTF-8
LF
Python
3.10.12 64-bit
4426k - 100% - 1.00
60%
Friday May 31, 3:32 PM


main.py - Assignment 1 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
ASSIGNMENT 1
__pycache__
Bilal
Wahaj
20k0208ai1.pdf
AI-ASG-1-23.pdf
Nqueens.cpp
nQueens.py
searches.cpp
AI-ASG-1-24.pdf
distances.txt
drive-download-20240218T172552Z-001.zip
heuristics.txt
main.py
Search.py
tempCodeRunnerFile.py
testing.py
OUTLINE
TIMELINE
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Code
python -u "/home/owaisk4/Win_backup/FAST NU assignments/Artificial Intelligence/Assignments/Assignment 1/main.py"
→ Assignment 1 python -u "/home/owaisk4/Win_backup/FAST NU assignments/Artificial Intelligence/Assignments/Assignment 1/main.py"
{'Fagaras', 'Vaslui', 'Drobeta', 'Pitesti', 'Timisoara', 'Craiova', 'Zerind', 'Bucharest', 'Urziceni', 'Mehadia', 'Arad', 'Neamt', 'Sibiu', 'Eforie', 'Rimnicu Vilcea', 'Hirsova', 'Giurgiu', 'Oradea', 'Lugoj', 'Iasi'}
Enter a source location from the above map: Fagaras
Enter a destination location from the above map: Lugoj
Using BreadthFirstSearch:
Path = Fagaras -> Sibiu -> Bucharest -> Oradea -> Arad -> Rimnicu Vilcea -> Pitesti -> Giurgiu -> Urziceni -> Zerind -> Timisoara -> Craiova -> Vaslui -> Hirsova -> Lugoj
Cost = 468
Using UniformCostSearch:
Path = Fagaras -> Sibiu -> Rimnicu Vilcea -> Bucharest -> Arad -> Oradea -> Pitesti -> Urziceni -> Giurgiu -> Zerind -> Craiova -> Timisoara -> Lugoj
Cost = 468
Using GreedyBestFirstSearch:
Path = None
Cost = None (Heuristics)
Using IterativeDeepeningDepthFirstSearch:
Path = Fagaras -> Sibiu -> Arad -> Timisoara -> Lugoj
Cost = 468
Enter a source location from the above map:
Launchpad
Run Testcases
Live Share
Ln 42, Col 1
Spaces: 4
UTF-8
LF
Python
3.10.12 64-bit
4426k - 100% - 1.00
60%
Friday May 31, 3:32 PM