

```
55.pv - C:/Users/HP/55.pv (3.4.3)
File Edit Format Run Options Window Help
class DSU:
   def union(self, x, y):
                                                                                        Python 3.4.3 Shell
                                                                                                                                                             ×
       ir xr == yr:
                                                                                        File Edit Shell Debug Options Window Help
           return False
       if self.rank[xr] < self.rank[yr]:
                                                                                        Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:44:40) [MSC v.1600 64 bit (AM
           self.parent[xr] = yr
                                                                                        D64)] on win32
       elif self.rank[xr] > self.rank[yr]:
                                                                                        Type "copyright", "credits" or "license()" for more information.
           self.parent[yr] = xr
                                                                                        else:
           self.parent[yr] = xr
                                                                                        Test Case 1 Output:
           self.rank[xr] += 1
                                                                                        Edges in MST: [(2, 3, 4), (0, 3, 5), (0, 1, 10)]
       return True
                                                                                        Total weight of MST: 19
# Kruskal's Algorithm
                                                                                        Test Case 2 Output:
def kruskal(n, edges):
                                                                                        Edges in MST: [(0, 1, 2), (1, 2, 3), (1, 4, 5), (0, 3, 6)]
   # Sort edges by weight
                                                                                        Total weight of MST: 16
   edges.sort(key=lambda x: x[2])
   dsu = DSU(n)
   mst = []
   total weight = 0
   for u, v, w in edges:
       if dsu.union(u, v):
           mst.append((u, v, w))
           total weight += w
   return mst, total weight
# Test Case 1
nl, ml = 4, 5
edges1 = [(0, 1, 10), (0, 2, 6), (0, 3, 5), (1, 3, 15), (2, 3, 4)]
mstl, weightl = kruskal(nl, edgesl)
print("Test Case 1 Output:")
print ("Edges in MST:", mstl)
print ("Total weight of MST:", weightl)
# Expected: [(2, 3, 4), (0, 3, 5), (0, 1, 10)], weight = 19
# Test Case 2
n2, m2 = 5, 7
edges2 = [(0, 1, 2), (0, 3, 6), (1, 2, 3), (1, 3, 8), (1, 4, 5), (2, 4, 7), (3, 4, 9)]
mst2, weight2 = kruskal(n2, edges2)
print("\nTest Case 2 Output:")
print ("Edges in MST:", mst2)
print("Total weight of MST:", weight2)
# Expected: [(0, 1, 2), (1, 2, 3), (1, 4, 5), (0, 3, 6)], weight = 16
                                                                                                                                                              Ln: 12 Col: 4
                                                                                                                                                                            Ln: 57 Col: 0
 NEP - WI
```

27-09-2025

Q Search

In 4 hours

```
op0.pv - C:/Users/HP/op0.pv (3.4.3)
File Edit Format Run Options Window Help
    while i < len(edges):
        for u, v, w in same weight edges:
            if dsu.find(u) != dsu.find(v):
                                                                                                   Python 3.4.3 Shell
                                                                                                                                                                           ×
       # If more than one possible edge could be chosen, uniqueness is broken
       if len(chosen edges) > 1:
                                                                                                   File Edit Shell Debug Options Window Help
            unique = False
                                                                                                   Python 3.4.3 (v3.4.3:9b73flc3e601, Feb 24 2015, 22:44:40) [MSC v.1600 64 bit (AM
       # Add chosen edges to MST
                                                                                                   Type "copyright", "credits" or "license()" for more information.
       for u, v, w in chosen edges:
                                                                                                   if dsu.union(u, v):
                                                                                                   >>>
               mst.append((u, v, w))
                                                                                                   Test Case 1 Output:
               total weight += w
                                                                                                   Is the given MST unique? True
    return mst, total weight, unique
                                                                                                   Test Case 2 Output:
                                                                                                   Is the given MST unique? False
def verify mst(n, edges, given mst):
                                                                                                   Another possible MST: [(0, 1, 1), (0, 2, 1), (1, 3, 2), (3, 4, 3)]
    mst, weight, unique = kruskal with uniqueness(n, edges)
                                                                                                   Total weight of MST: 7
    given weight = sum(w for , , w in given mst)
                                                                                                   >>>
    if given weight != weight:
       return False, None, None # given MST invalid
    if unique:
       return True, None, weight
       return False, mst, weight
# Test Case 1
nl. ml = 4.5
edges1 = [(0, 1, 10), (0, 2, 6), (0, 3, 5), (1, 3, 15), (2, 3, 4)]
given mst1 = [(2, 3, 4), (0, 3, 5), (0, 1, 10)]
uniquel, alt1, w1 = verify mst(n1, edges1, given mst1)
print("Test Case 1 Output:")
print ("Is the given MST unique?", uniquel)
# Test Case 2
n2, m2 = 5, 6
edges2 = [(0, 1, 1), (0, 2, 1), (1, 3, 2), (2, 3, 2), (3, 4, 3), (4, 2, 3)]
given mst2 = [(0, 1, 1), (0, 2, 1), (1, 3, 2), (3, 4, 3)]
unique2, alt2, w2 = verify mst(n2, edges2, given mst2)
print("\nTest Case 2 Output:")
print("Is the given MST unique?", unique2)
if not unique2:
    print ("Another possible MST:", alt2)
   print ("Total weight of MST:", w2)
                                                                                                                                                                            Ln: 12 Col: 4
                                                                                                                                                                                   Ln: 86 Col: 0
```













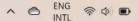






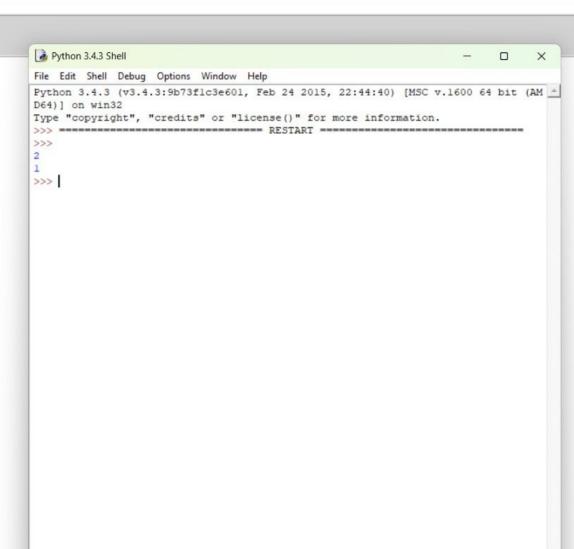


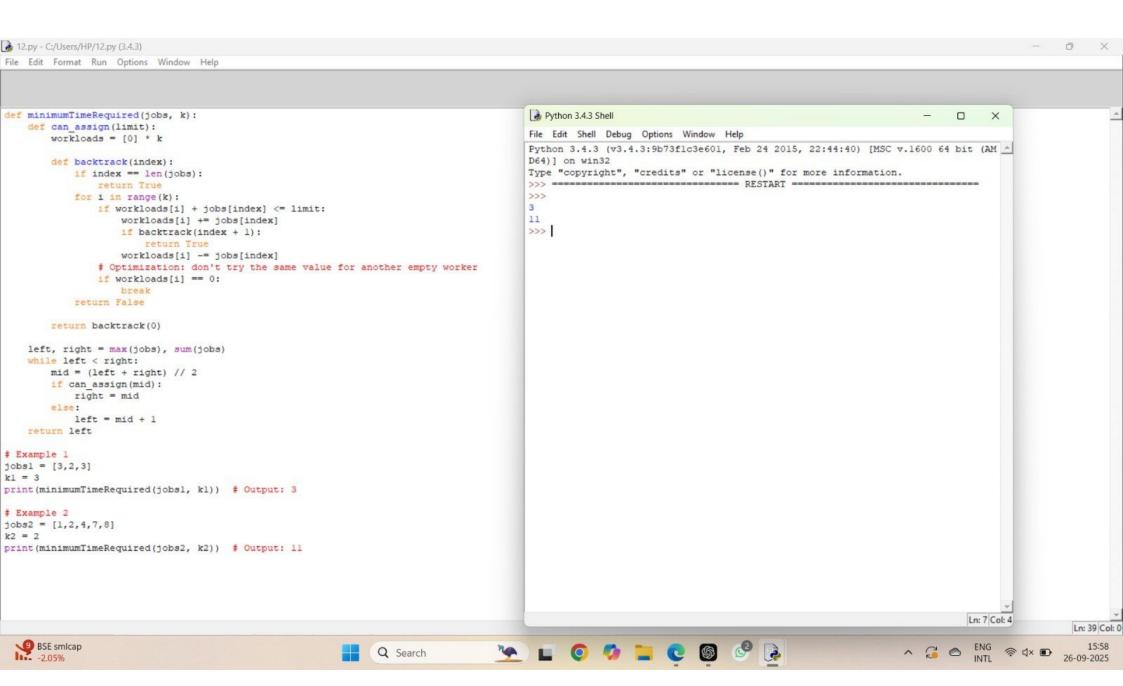


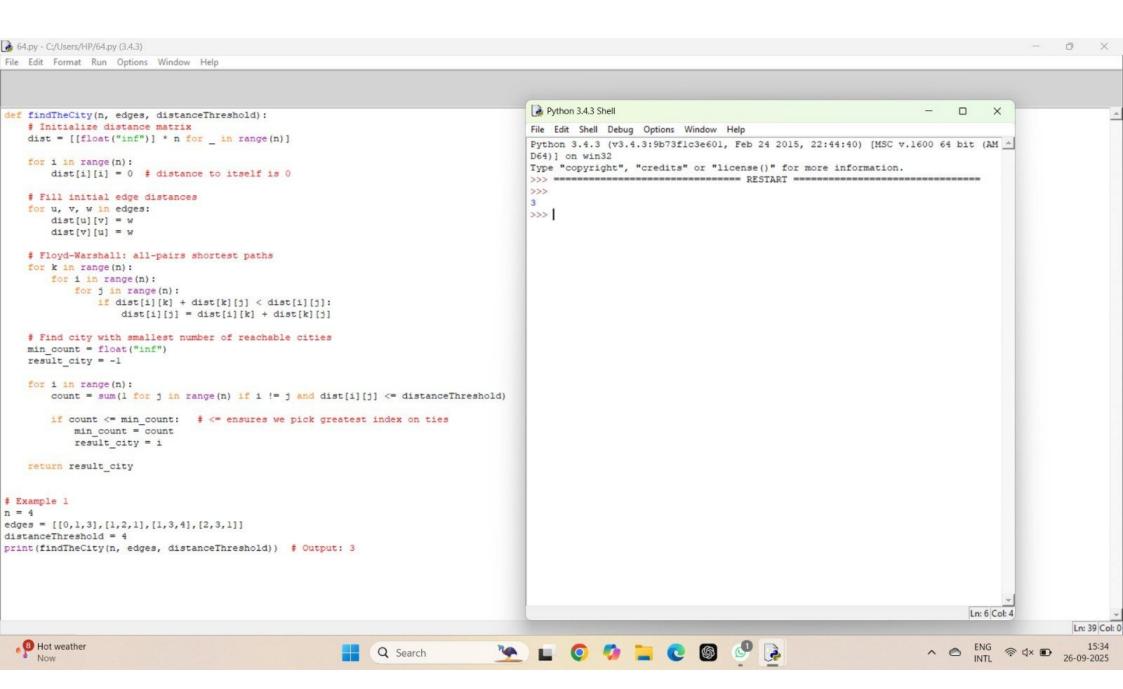


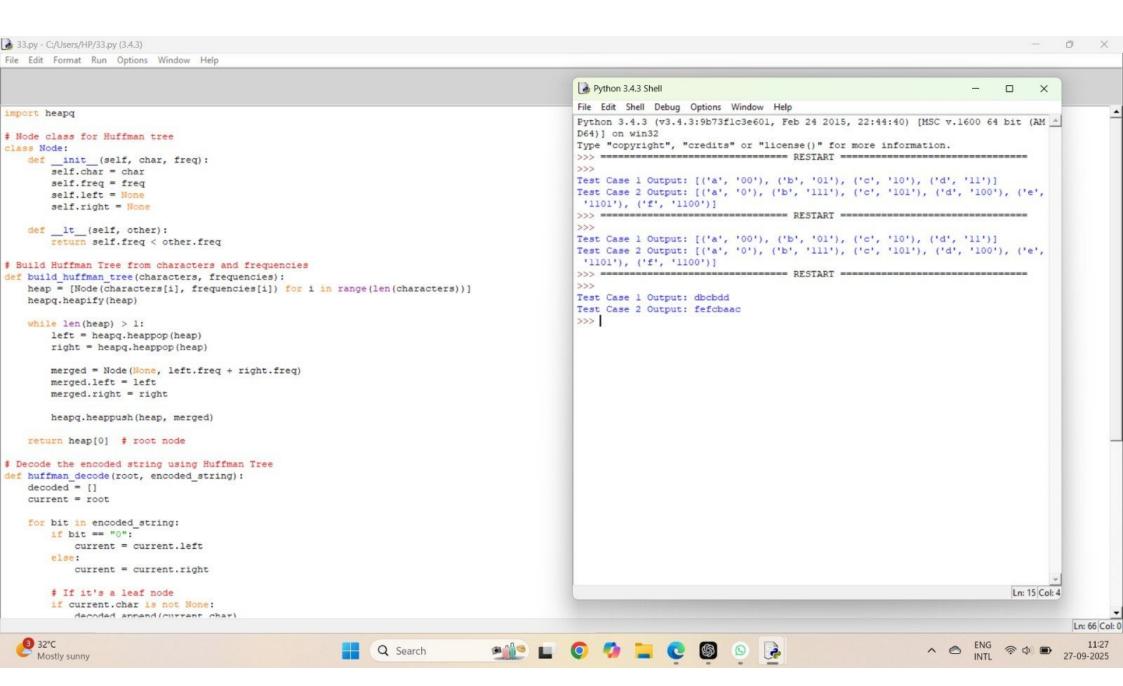


```
def min coins to add(coins, target):
   coins.sort() # Sort coins to process them in increasing order
   added coins = 0
   reach = 0 # Maximum sum we can achieve with current coins
   i = 0
   while reach < target:
       if i < len(coins) and coins[i] <= reach + 1:
           # If the current coin can extend the reach, use it
           reach += coins[i]
           i += 1
       else:
           # Otherwise, we need to add a coin of value reach+1
           added coins += 1
           reach += reach + 1 # Extend the reach optimally by adding reach+1
   return added coins
# Example 1
coins1 = [1, 4, 10]
target1 = 19
print(min coins to add(coinsl, targetl)) # Output: 2
# Example 2
coins2 = [1, 4, 10, 5, 7, 19]
target2 = 19
print(min_coins_to_add(coins2, target2)) # Output: 1
```









```
op0.py - C:/Users/HP/op0.py (3.4.3)
File Edit Format Run Options Window Help
import matplotlib.pyplot as plt
                                                                                           Python 3.4.3 Shell
                                                                                                                                                                 ×
                                                                                           File Edit Shell Debug Options Window Help
def solve_n_queens(n, max_solutions=None):
                                                                                           Python 3.4.3 (v3.4.3:9b73flc3e601, Feb 24 2015, 22:44:40) [MSC v.1600 64 bit (AM
   solutions = []
   cols = set()
                                                                                           Type "copyright", "credits" or "license()" for more information.
   diagl = set() # r - c
   diag2 = set() # r + c
                                                                                           >>>
   board = [-1] * n # board[r] = c position of queen in row r
                                                                                           Test Case 1 Output:
   def backtrack(r):
                                                                                           Is the given MST unique? True
       if max solutions is not None and len(solutions) >= max solutions:
                                                                                           Test Case 2 Output:
                                                                                           Is the given MST unique? False
       if r == n:
                                                                                           Another possible MST: [(0, 1, 1), (0, 2, 1), (1, 3, 2), (3, 4, 3)]
           solutions.append(board.copy())
                                                                                           Total weight of MST: 7
                                                                                                          RESTART -----
       for c in range(n):
                                                                                           >>>
           if c in cols or (r - c) in diagl or (r + c) in diag2:
                                                                                           Traceback (most recent call last):
           cols.add(c); diagl.add(r - c); diag2.add(r + c)
                                                                                            File "C:/Users/HP/op0.py", line 1, in <module>
                                                                                              import matplotlib.pyplot as plt
          board[r] = c
                                                                                           ImportError: No module named 'matplotlib'
          backtrack(r + 1)
           cols.remove(c); diagl.remove(r - c); diagl.remove(r + c)
          board[r] = -1
   backtrack(0)
   return solutions
def board to text (board):
   n = len(board)
   rows = []
   for r in range(n):
       row = ['.'] * n
       c = board[r]
       row[c] = 'Q'
       rows.append(''.join(row))
   return '\n'.join(rows)
def plot board(board, title=None, show=True):
   n = len(board)
   # construct matrix with 1 wh
                                                                                                                                                              Ln: 18 Col: 4
                                                                                                                                                                     Ln: 41 Col: 0
                                                         Hot days ahead 34°C
```

