33.py - C:/Users/HP/33.py (3.4.3)

File Edit Format Run Options Window Help

```python
def min_containers(weights, max_capacity):
    # Sort items in descending order (heaviest first)
    weights.sort(reverse=True)

    containers = []

    for w in weights:
        placed = False
        # Try to place item in an existing container
        for i in range(len(containers)):
            if containers[i] + w <= max_capacity:
                containers[i] += w
                placed = True
                break
        # If it doesn't fit in any container, create a new one
        if not placed:
            containers.append(w)

    return len(containers)

# Test Case 1
weights1 = [5, 10, 15, 20, 25, 30, 35]
max_capacity1 = 50
print("Test Case 1 Output:", min_containers(weights1, max_capacity1))  # Expected: 4

# Test Case 2
weights2 = [10, 20, 30, 40, 50, 60, 70, 80]
max_capacity2 = 100
print("Test Case 2 Output:", min_containers(weights2, max_capacity2))  # Expected: 6
```

Python 3.4.3 Shell

File Edit Shell Debug Options Window Help

```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:44:40) [MSC v.1600 64 bit (AM
D64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
Test Case 1 Output: 60
Test Case 2 Output: 50
>>> ================================ RESTART ================================
>>>
Test Case 1 Output: 3
Test Case 2 Output: 4
>>>
```

Ln: 11 Col: 4

Ln: 30 Col: 0

Breaking news
Speeding Thar s...

Search

ENG
INTL

11:30
27-09-2025

55.py - C:/Users/HP/55.py (3.4.3)

File Edit Format Run Options Window Help

```python
class DSU:
    def union(self, x, y):
        if xr == yr:
            return False
        if self.rank[xr] < self.rank[yr]:
            self.parent[xr] = yr
        elif self.rank[xr] > self.rank[yr]:
            self.parent[yr] = xr
        else:
            self.parent[yr] = xr
            self.rank[xr] += 1
        return True

# Kruskal's Algorithm
def kruskal(n, edges):
    # Sort edges by weight
    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
n1, m1 = 4, 5
edges1 = [(0, 1, 10), (0, 2, 6), (0, 3, 5), (1, 3, 15), (2, 3, 4)]
mst1, weight1 = kruskal(n1, edges1)
print("Test Case 1 Output:")
print("Edges in MST:", mst1)
print("Total weight of MST:", weight1)
# 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
```

Python 3.4.3 Shell

File Edit Shell Debug Options Window Help

```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:44:40) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
Test Case 1 Output:
Edges in MST: [(2, 3, 4), (0, 3, 5), (0, 1, 10)]
Total weight of MST: 19

Test Case 2 Output:
Edges in MST: [(0, 1, 2), (1, 2, 3), (1, 4, 5), (0, 3, 6)]
Total weight of MST: 16
>>>
```

Ln: 12 Col: 4

Ln: 57 Col: 0

NEP - WI In 4 hours Search ENG INTL 15:53 27-09-2025 op0.py - C:/Users/HP/op0.py (3.4.3)

File Edit Format Run Options Window Help

```python
    while i < len(edges):
        for u, v, w in same_weight_edges:
            if dsu.find(u) != dsu.find(v):

        # If more than one possible edge could be chosen, uniqueness is broken
        if len(chosen_edges) > 1:
            unique = False

        # Add chosen edges to MST
        for u, v, w in chosen_edges:
            if dsu.union(u, v):
                mst.append((u, v, w))
                total_weight += w

    return mst, total_weight, unique

def verify_mst(n, edges, given_mst):
    mst, weight, unique = kruskal_with_uniqueness(n, edges)
    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
    else:
        return False, mst, weight

# Test Case 1
n1, m1 = 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)]
unique1, alt1, w1 = verify_mst(n1, edges1, given_mst1)
print("Test Case 1 Output:")
print("Is the given MST unique?", unique1)

# 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)
```

Python 3.4.3 Shell

File Edit Shell Debug Options Window Help

```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:44:40) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
Test Case 1 Output:
Is the given MST unique? True

Test Case 2 Output:
Is the given MST unique? False
Another possible MST: [(0, 1, 1), (0, 2, 1), (1, 3, 2), (3, 4, 3)]
Total weight of MST: 7
>>>
```

Ln: 12 Col: 4

Ln: 86 Col: 0

35°C Partly sunny Search ENG INTL 15:59 27-09-2025

File Edit Format Run Options Window Help

```python
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(coins1, target1))  # Output: 2

# Example 2
coins2 = [1, 4, 10, 5, 7, 19]
target2 = 19
print(min_coins_to_add(coins2, target2))  # Output: 1
```

Python 3.4.3 Shell

File Edit Shell Debug Options Window Help

```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:44:40) [MSC v.1600 64 bit (AM
D64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
2
1
>>>
```

12.py - C:/Users/HP/12.py (3.4.3)

```python
def minimumTimeRequired(jobs, k):
    def can_assign(limit):
        workloads = [0] * k

        def backtrack(index):
            if index == len(jobs):
                return True
            for i in range(k):
                if workloads[i] + jobs[index] <= limit:
                    workloads[i] += jobs[index]
                    if backtrack(index + 1):
                        return True
                    workloads[i] -= jobs[index]
                # Optimization: don't try the same value for another empty worker
                if workloads[i] == 0:
                    break
            return False

        return backtrack(0)

    left, right = max(jobs), sum(jobs)
    while left < right:
        mid = (left + right) // 2
        if can_assign(mid):
            right = mid
        else:
            left = mid + 1
    return left

# Example 1
jobs1 = [3,2,3]
k1 = 3
print(minimumTimeRequired(jobs1, k1))  # Output: 3

# Example 2
jobs2 = [1,2,4,7,8]
k2 = 2
print(minimumTimeRequired(jobs2, k2))  # Output: 11
```

Python 3.4.3 Shell

```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:44:40) [MSC v.1600 64 bit (AM
D64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
3
11
>>>
```

64.py - C:/Users/HP/64.py (3.4.3)

File Edit Format Run Options Window Help

```python
def findTheCity(n, edges, distanceThreshold):
    # Initialize distance matrix
    dist = [[float("inf")] * n for _ in range(n)]

    for i in range(n):
        dist[i][i] = 0  # distance to itself is 0

    # Fill initial edge distances
    for u, v, w in edges:
        dist[u][v] = w
        dist[v][u] = w

    # Floyd-Warshall: all-pairs shortest paths
    for k in range(n):
        for i in range(n):
            for j in range(n):
                if dist[i][k] + dist[k][j] < dist[i][j]:
                    dist[i][j] = dist[i][k] + dist[k][j]

    # Find city with smallest number of reachable cities
    min_count = float("inf")
    result_city = -1

    for i in range(n):
        count = sum(1 for j in range(n) if i != j and dist[i][j] <= distanceThreshold)

        if count <= min_count:   # <= ensures we pick greatest index on ties
            min_count = count
            result_city = i

    return result_city


# Example 1
n = 4
edges = [[0,1,3],[1,2,1],[1,3,4],[2,3,1]]
distanceThreshold = 4
print(findTheCity(n, edges, distanceThreshold))  # Output: 3
```

Python 3.4.3 Shell

File Edit Shell Debug Options Window Help

```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:44:40) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
3
>>>
```

Ln: 6 Col: 4

Ln: 39 Col: 0

Hot weather
Now

Search

ENG
INTL

15:34
26-09-2025

33.py - C:/Users/HP/33.py (3.4.3)

File Edit Format Run Options Window Help

```python
import heapq

# Node class for Huffman tree
class Node:
    def __init__(self, char, freq):
        self.char = char
        self.freq = freq
        self.left = None
        self.right = None

    def __lt__(self, other):
        return self.freq < other.freq

# Build Huffman Tree from characters and frequencies
def build_huffman_tree(characters, frequencies):
    heap = [Node(characters[i], frequencies[i]) for i in range(len(characters))]
    heapq.heapify(heap)

    while len(heap) > 1:
        left = heapq.heappop(heap)
        right = heapq.heappop(heap)

        merged = Node(None, left.freq + right.freq)
        merged.left = left
        merged.right = right

        heapq.heappush(heap, merged)

    return heap[0]  # root node

# Decode the encoded string using Huffman Tree
def huffman_decode(root, encoded_string):
    decoded = []
    current = root

    for bit in encoded_string:
        if bit == "0":
            current = current.left
        else:
            current = current.right

        # If it's a leaf node
        if current.char is not None:
            decoded.append(current.char)
```

Ln: 66 Col: 0

Python 3.4.3 Shell

File Edit Shell Debug Options Window Help

```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:44:40) [MSC v.1600 64 bit (AM
D64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
Test Case 1 Output: [('a', '00'), ('b', '01'), ('c', '10'), ('d', '11')]
Test Case 2 Output: [('a', '0'), ('b', '111'), ('c', '101'), ('d', '100'), ('e',
 '1101'), ('f', '1100')]
>>> ================================ RESTART ================================
>>>
Test Case 1 Output: [('a', '00'), ('b', '01'), ('c', '10'), ('d', '11')]
Test Case 2 Output: [('a', '0'), ('b', '111'), ('c', '101'), ('d', '100'), ('e',
 '1101'), ('f', '1100')]
>>> ================================ RESTART ================================
>>>
Test Case 1 Output: dbcbdd
Test Case 2 Output: fefcbaac
>>>
```

Ln: 15 Col: 4

32°C Mostly sunny

Search

ENG INTL 11:27 27-09-2025

op0.py - C:/Users/HP/op0.py (3.4.3)

File Edit Format Run Options Window Help

```python
import matplotlib.pyplot as plt

def solve_n_queens(n, max_solutions=None):
    solutions = []
    cols = set()
    diag1 = set()  # r - c
    diag2 = set()  # r + c
    board = [-1] * n  # board[r] = c position of queen in row r

    def backtrack(r):
        if max_solutions is not None and len(solutions) >= max_solutions:
            return
        if r == n:
            solutions.append(board.copy())
            return
        for c in range(n):
            if c in cols or (r - c) in diag1 or (r + c) in diag2:
                continue
            cols.add(c); diag1.add(r - c); diag2.add(r + c)
            board[r] = c
            backtrack(r + 1)
            cols.remove(c); diag1.remove(r - c); diag2.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
```

Python 3.4.3 Shell

File Edit Shell Debug Options Window Help

```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:44:40) [MSC v.1600 64 bit (AM
D64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
Test Case 1 Output:
Is the given MST unique? True

Test Case 2 Output:
Is the given MST unique? False
Another possible MST: [(0, 1, 1), (0, 2, 1), (1, 3, 2), (3, 4, 3)]
Total weight of MST: 7
>>> ================================ RESTART ================================
>>>
Traceback (most recent call last):
  File "C:/Users/HP/op0.py", line 1, in <module>
    import matplotlib.pyplot as plt
ImportError: No module named 'matplotlib'
>>>
```

Ln: 18 Col: 4

Ln: 41 Col: 0

Hot days ahead 34°C Search ENG INTL 16:10 27-09-2025

64.py - C:/Users/HP/64.py (3.4.3)

File Edit Format Run Options Window Help

```python
import heapq
from collections import defaultdict

def networkDelayTime(times, n, k):
    # Build adjacency list
    graph = defaultdict(list)
    for u, v, w in times:
        graph[u].append((v, w))

    # Min-heap for Dijkstra (time, node)
    heap = [(0, k)]
    dist = {}

    while heap:
        time, node = heapq.heappop(heap)
        if node in dist:
            continue
        dist[node] = time
        for nei, w in graph[node]:
            if nei not in dist:
                heapq.heappush(heap, (time + w, nei))

    if len(dist) == n:
        return max(dist.values())
    return -1


# Example 1
times = [[2,1,1],[2,3,1],[3,4,1]]
print(networkDelayTime(times, 4, 2))  # Output: 2

# Example 2
times = [[1,2,1]]
print(networkDelayTime(times, 2, 1))  # Output: 1

# Example 3
times = [[1,2,1]]
print(networkDelayTime(times, 2, 2))  # Output: -1
```

Python 3.4.3 Shell

File Edit Shell Debug Options Window Help

```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:44:40) [MSC v.1600 64 bit (AM
D64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
2
1
-1
>>>
```

Ln: 8 Col: 4

Ln: 39 Col: 0

BSE midcap -1.95% Search ENG INTL 15:43 26-09-2025

64.py - C:/Users/HP/64.py (3.4.3)

```python
from collections import Counter

def numIdenticalPairs(nums):
    freq = Counter(nums)
    count = 0
    for k in freq.values():
        count += k * (k - 1) // 2
    return count


# Example 1
print(numIdenticalPairs([1,2,3,1,1,3]))  # Output: 4

# Example 2
print(numIdenticalPairs([1,1,1,1]))      # Output: 6
```

Python 3.4.3 Shell

```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:44:40) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
4
6
>>>
```