

```
1 def maxCoins(piles):
2     piles.sort(reverse=True)
3     return sum(piles[i] for i in range(1, len(piles), 2))
4
5 # Example usage:
6 print(maxCoins([2,4,1,2,7,8])) # Output: 9
7 print(maxCoins([2,4,5])) # Output: 4
8
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

[Running] python -u "c:\Users\hp\OneDrive\Desktop\project directory\tempCodeRunnerFile.python"

10

4

[Done] exited with code=0 in 0.161 seconds

```
1  def minCoins(coins, target):
2      coins.sort()
3      missing, count = 1, 0
4      for coin in coins:
5          while missing < coin:
6              missing += missing
7              count += 1
8          missing += coin
9      while missing <= target:
10         missing += missing
11         count += 1
12     return count
13
14 # Example usage:
15 print(minCoins([1,4,10], 19)) # Output: 2
16 print(minCoins([1, 4, 10, 5, 7, 19], 19)) # Output: 1
17
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Code

[Running] python -u "c:\Users\hp\OneDrive\Desktop\project directory\tempCodeRunnerFile.python"

2

1

[Done] exited with code=0 in 0.146 seconds

```

1  def minimumTimeRequired(jobs, k):
2      def canFinish(limit):
3          workers = [0] * k
4          jobs.sort(reverse=True)
5
6          def dfs(i):
7              if i == len(jobs):
8                  return True
9              for j in range(k):
10                 if workers[j] + jobs[i] <= limit:
11                     workers[j] += jobs[i]
12                     if dfs(i + 1):
13                         return True
14                     workers[j] -= jobs[i]
15                 if workers[j] == 0:
16                     break
17             return False
18
19         return dfs(0)
20
21     l, r = max(jobs), sum(jobs)
22     while l < r:
23         mid = (l + r) // 2
24         if canFinish(mid):
25             r = mid
26         else:
27             l = mid + 1
28     return l

```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Code



[Running] python -u "c:\Users\hp\OneDrive\Desktop\project directory\tempCodeRunnerFile.python"

2

1

[Done] exited with code=0 in 0.146 seconds

```
def minimumTimeRequired(jobs, k): • Untitled-4
```

```
def jobScheduling(startTime, endTime, profit):
    jobs = sorted(zip(startTime, endTime, profit), key=lambda x: x[1])
    dp = [(0, 0)] # (end time, profit)

    for start, end, p in jobs:
        i = bisect_right(dp, (start, float('inf'))) - 1
        if dp[i][1] + p > dp[-1][1]:
            dp.append((end, dp[i][1] + p))

    return dp[-1][1]

# Example usage:
print(jobScheduling([1,2,3,3], [3,4,5,6], [50,10,40,70])) # Output: 120
print(jobScheduling([1,2,3,4,6], [3,5,10,6,9], [20,20,100,70,60])) # Output: 150
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Code



```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\project
directory\tempCodeRunnerFile.python"
```

```
120
```

```
150
```

```
[Done] exited with code=0 in 0.202 seconds
```

```

1  import heapq
2
3  def dijkstra_matrix(n, graph, source):
4      distances = [float('inf')] * n
5      distances[source] = 0
6      pq = [(0, source)]
7
8      while pq:
9          current_distance, u = heapq.heappop(pq)
10         if current_distance > distances[u]:
11             continue
12         for v in range(n):
13             if graph[u][v] != float('inf'):
14                 distance = current_distance + graph[u][v]
15                 if distance < distances[v]:
16                     distances[v] = distance
17                     heapq.heappush(pq, (distance, v))
18     return distances
19
20 # Example usage:
21 print(dijkstra_matrix(5, [[0, 10, 3, float('inf'), float('inf')], [float('inf'), 0, 1, 2, float('inf')],
22                             [float('inf'), 4, 0, 8, 2], [float('inf'), float('inf'), float('inf'), 0, 7],
23                             [float('inf'), float('inf'), float('inf'), 9, 0]], 0)) # Output: [0, 7, 3, 9, 5]
24

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[Done] exited with code=0 in 0.146 seconds

[Running] python -u "c:\Users\hp\OneDrive\Desktop\project directory\tempCodeRunnerFile.python"

120

150

[Done] exited with code=0 in 0.202 seconds

```

1  import heapq
2  def dijkstra_edges(n, edges, source, target):
3      graph = [[] for _ in range(n)]
4      for u, v, w in edges:
5          graph[u].append((v, w))
6          graph[v].append((u, w))
7
8      distances = [float('inf')] * n
9      distances[source] = 0
10     pq = [(0, source)]
11
12     while pq:
13         current_distance, u = heapq.heappop(pq)
14         if u == target:
15             return current_distance
16         if current_distance > distances[u]:
17             continue
18         for v, weight in graph[u]:
19             distance = current_distance + weight
20             if distance < distances[v]:
21                 distances[v] = distance
22                 heapq.heappush(pq, (distance, v))
23
24     return distances[target]
25
26 # Example usage:
27 print(dijkstra_edges(6, [(0, 1, 7), (0, 2, 9), (0, 5, 14), (1, 2, 10), (1, 3, 15),
28                          (2, 3, 11), (2, 5, 2), (3, 4, 6), (4, 5, 9)], 0, 4)) # Output: 20

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[Done] exited with code=1 in 0.145 seconds

[Running] python -u "c:\Users\hp\OneDrive\Desktop\project directory\tempCodeRunnerFile.python"
20

[Done] exited with code=0 in 0.147 seconds

```

1  import heapq
2  from collections import defaultdict
3
4  def huffmanCoding(characters, frequencies):
5      heap = [[weight, [char, ""]] for char, weight in zip(characters, frequencies)]
6      heapq.heapify(heap)
7
8      while len(heap) > 1:
9          lo = heapq.heappop(heap)
10         hi = heapq.heappop(heap)
11         for pair in lo[1:]:
12             pair[1] = '0' + pair[1]
13         for pair in hi[1:]:
14             pair[1] = '1' + pair[1]
15         heapq.heappush(heap, [lo[0] + hi[0]] + lo[1:] + hi[1:])
16
17     return sorted(heap[0][1:], key=lambda x: (len(x[-1]), x))
18
19 # Example usage:
20 print(huffmanCoding(['a', 'b', 'c', 'd'], [5, 9, 12, 13]))
21

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[Done] exited with code=0 in 0.147 seconds

[Running] python -u "c:\Users\hp\OneDrive\Desktop\project directory\tempCodeRunnerFile.python"
 [['a', '00'], ['b', '01'], ['c', '10'], ['d', '11']]

[Done] exited with code=0 in 0.152 seconds

```

1  import heapq
2  class TreeNode:
3      def __init__(self, char=None, freq=None):
4          self.char, self.freq = char, freq
5          self.left = self.right = None
6      def __lt__(self, other):
7          return self.freq < other.freq
8  def build_huffman_tree(chars, freqs):
9      heap = [TreeNode(c, f) for c, f in zip(chars, freqs)]
10     heapq.heapify(heap)
11     while len(heap) > 1:
12         left, right = heapq.heappop(heap), heapq.heappop(heap)
13         parent = TreeNode(freq=left.freq + right.freq)
14         parent.left, parent.right = left, right
15         heapq.heappush(heap, parent)
16     return heap[0]
17 def huffman_decode(encoded, chars, freqs):
18     root = build_huffman_tree(chars, freqs)
19     decoded, node = "", root
20     for bit in encoded:
21         node = node.left if bit == '0' else node.right
22         if node.char:
23             decoded += node.char
24         node = root
25     return decoded
26 chars = ['a', 'b', 'c', 'd']
27 freqs = [5, 9, 12, 13]
28 encoded_str = '1101100111110'
29 print(huffman_decode(encoded_str, chars, freqs)) # Output: "abacd"
30

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[Done] exited with code=1 in 0.151 seconds

[Running] python -u "c:\Users\hp\OneDrive\Desktop\project directory\tempCodeRunnerFile.python"
dbcbdd

[Done] exited with code=0 in 0.147 seconds


```
1 def maxWeight(weights, max_capacity):
2     weights.sort(reverse=True)
3     current_weight = 0
4     for weight in weights:
5         if current_weight + weight <= max_capacity:
6             current_weight += weight
7     return current_weight
8
9 # Example usage:
10 print(maxWeight([10, 20, 30, 40, 50], 60))
11
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[Running] python -u "c:\Users\hp\OneDrive\Desktop\project directory\tempCodeRunnerFile.python"
60

[Done] exited with code=0 in 0.124 seconds

```
1 def minContainers(weights, max_capacity):
2     weights.sort(reverse=True)
3     containers = 0
4     while weights:
5         current_weight = 0
6         i = 0
7         while i < len(weights):
8             if current_weight + weights[i] <= max_capacity:
9                 current_weight += weights.pop(i)
10            else:
11                i += 1
12            containers += 1
13     return containers
14
15 # Example usage:
16 print(minContainers([5, 10, 15, 20, 25, 30, 35], 50)) # Output: 4
17 |
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[Running] python -u "c:\Users\hp\OneDrive\Desktop\project directory\tempCodeRunnerFile.python"
3

[Done] exited with code=0 in 0.155 seconds

```

1 def kruskalMST(n, edges):
2     edges.sort(key=lambda x: x[2])
3     parent = list(range(n))
4
5     def find(x):
6         if parent[x] != x:
7             parent[x] = find(parent[x])
8         return parent[x]
9
10    mst, total_weight = [], 0
11    for u, v, weight in edges:
12        root_u, root_v = find(u), find(v)
13        if root_u != root_v:
14            parent[root_u] = root_v
15            mst.append((u, v, weight))
16            total_weight += weight
17            if len(mst) == n - 1:
18                break
19
20    return mst, total_weight
21
22 # Example usage:
23 print(kruskalMST(4, [(0, 1, 10), (0, 2, 6), (0, 3, 5), (1, 3, 15), (2, 3, 4)]))
24

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[Running] python -u "c:\Users\hp\OneDrive\Desktop\project directory\tempCodeRunnerFile.python"
 ((2, 3, 4), (0, 3, 5), (0, 1, 10)), 19)

[Done] exited with code=0 in 0.143 seconds

```

1 class UnionFind:
2     def __init__(self, n):
3         self.parent = list(range(n))
4     def find(self, u):
5         if self.parent[u] != u:
6             self.parent[u] = self.find(self.parent[u])
7         return self.parent[u]
8     def union(self, u, v):
9         pu, pv = self.find(u), self.find(v)
10        if pu != pv:
11            self.parent[pu] = pv
12            return True
13        return False
14    def kruskal(n, edges):
15        uf, mst_edges, weight = UnionFind(n), [], 0
16        for u, v, w in sorted(edges, key=lambda x: x[2]):
17            if uf.union(u, v):
18                mst_edges.append((u, v, w))
19                weight += w
20                if len(mst_edges) == n - 1: break
21        return mst_edges, weight
22    def is_unique_mst(n, edges, given_mst):
23        mst_edges, mst_weight = kruskal(n, edges)
24        if sum(w for _, _, w in given_mst) != mst_weight:
25            return False, []
26        for u, v, w in edges:
27            uf, temp_weight, temp_mst = UnionFind(n), 0, []
28            for x, y, z in edges:
29                if uf.union(x, y) and (x, y, z) != (u, v, w):
30                    temp_mst.append((x, y, z))
31                    temp_weight += z
32            if len(temp_mst) == n - 1 and temp_weight == mst_weight and temp_mst != mst_edges:
33                return False, temp_mst
34        return True, []

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[Running] python -u C:\Users\inp\OneDrive\Desktop\project-directory\tempcode runner\file.py python
(True, [])