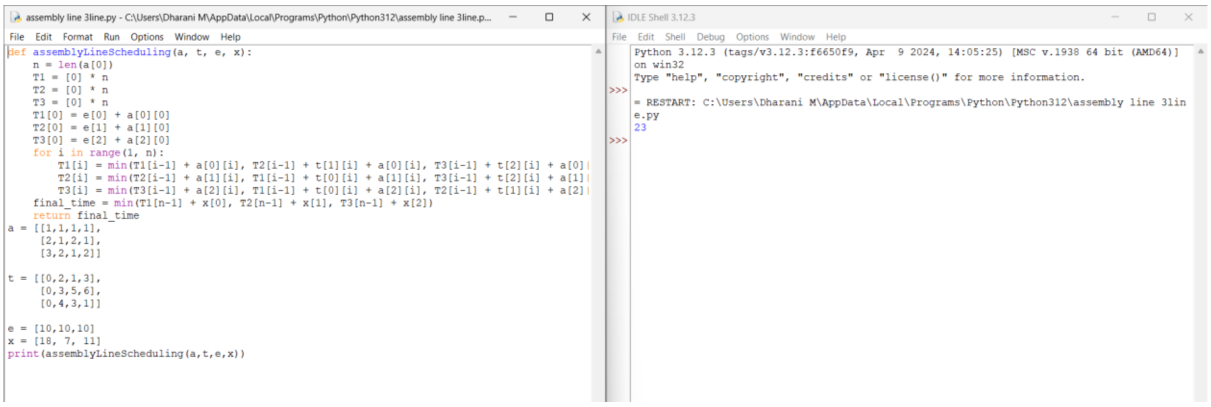
1. Assembly line


assembly line 3line.py - C:\Users\Dharani M\AppData\Local\Programs\Python\Python312\assembly line 3line.p...
File Edit Format Run Options Window Help
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help

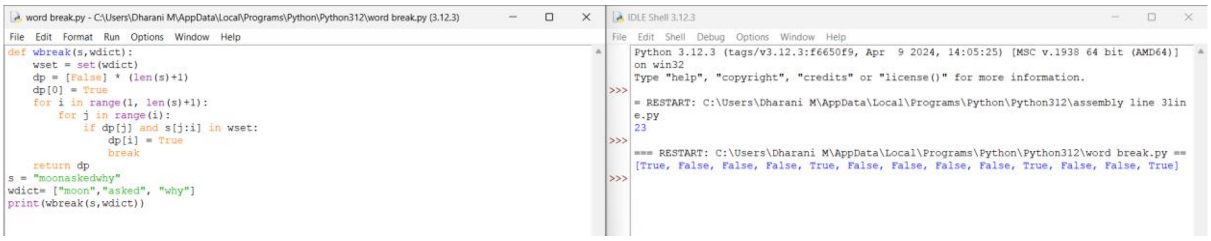
```
def assemblyLineScheduling(a, t, e, x):
    n = len(a[0])
    T1 = [0] * n
    T2 = [0] * n
    T3 = [0] * n
    T1[0] = e[0] + a[0][0]
    T2[0] = e[1] + a[1][0]
    T3[0] = e[2] + a[2][0]
    for i in range(1, n):
        T1[i] = min(T1[i-1] + a[0][i], T2[i-1] + t[1][i] + a[0][i], T3[i-1] + t[2][i] + a[0]
        T2[i] = min(T2[i-1] + a[1][i], T1[i-1] + t[0][i] + a[1][i], T3[i-1] + t[2][i] + a[1]
        T3[i] = min(T3[i-1] + a[2][i], T1[i-1] + t[0][i] + a[2][i], T2[i-1] + t[1][i] + a[2]
    final_time = min(T1[n-1] + x[0], T2[n-1] + x[1], T3[n-1] + x[2])
    return final_time
a = [[1,1,1,1],
     [2,1,2,1],
     [3,2,1,2]]

t = [[0,2,1,3],
     [0,3,5,6],
     [0,4,3,1]]

e = [10,10,10]
x = [18, 7, 11]
print(assemblyLineScheduling(a,t,e,x))
```

```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Dharani M\AppData\Local\Programs\Python\Python312\assembly line 3lin
e.py
23
>>>
```

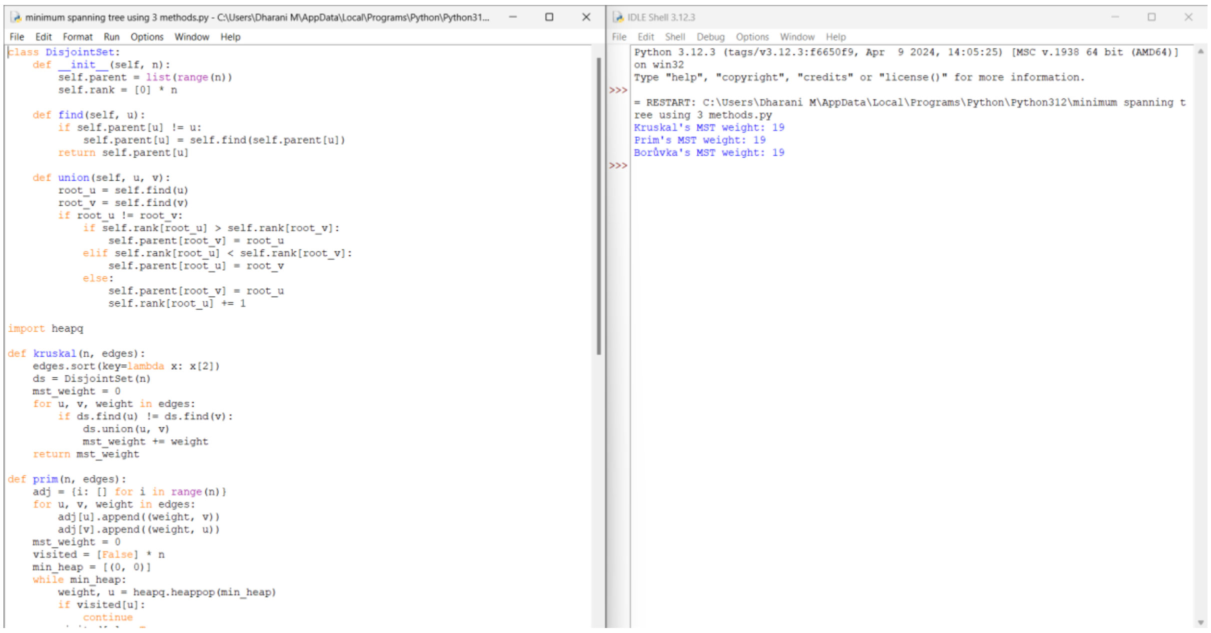
2. Word break


word break.py - C:\Users\Dharani M\AppData\Local\Programs\Python\Python312\word break.py (3.12.3)
File Edit Format Run Options Window Help
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help

```
def wbreak(s,wdict):
    wset = set(wdict)
    dp = [False] * (len(s)+1)
    dp[0] = True
    for i in range(1, len(s)+1):
        for j in range(i):
            if dp[j] and s[j:i] in wset:
                dp[i] = True
                break
    return dp
s = "moonaskedwhy"
wdict= ["moon","asked", "why"]
print(wbreak(s,wdict))
```

```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Dharani M\AppData\Local\Programs\Python\Python312\assembly line 3lin
e.py
23
>>>
=== RESTART: C:\Users\Dharani M\AppData\Local\Programs\Python\Python312\word break.py ==
[True, False, False, False, True, False, False, False, False, True, False, False, True]
>>>
```

3. Minimum spanning tree


minimum spanning tree using 3 methods.py - C:\Users\Dharani M\AppData\Local\Programs\Python\Python31...
File Edit Format Run Options Window Help
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help

```
class DisjointSet:
    def __init__(self, n):
        self.parent = list(range(n))
        self.rank = [0] * n

    def find(self, u):
        if self.parent[u] != u:
            self.parent[u] = self.find(self.parent[u])
        return self.parent[u]

    def union(self, u, v):
        root_u = self.find(u)
        root_v = self.find(v)
        if root_u != root_v:
            if self.rank[root_u] > self.rank[root_v]:
                self.parent[root_v] = root_u
            elif self.rank[root_u] < self.rank[root_v]:
                self.parent[root_u] = root_v
            else:
                self.parent[root_v] = root_u
                self.rank[root_u] += 1

import heapq

def kruskal(n, edges):
    edges.sort(key=lambda x: x[2])
    ds = DisjointSet(n)
    mst_weight = 0
    for u, v, weight in edges:
        if ds.find(u) != ds.find(v):
            ds.union(u, v)
            mst_weight += weight
    return mst_weight

def prim(n, edges):
    adj = {i: [] for i in range(n)}
    for u, v, weight in edges:
        adj[u].append((weight, v))
        adj[v].append((weight, u))
    mst_weight = 0
    visited = [False] * n
    min_heap = [(0, 0)]
    while min_heap:
        weight, u = heapq.heappop(min_heap)
        if visited[u]:
            continue
```

```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Dharani M\AppData\Local\Programs\Python\Python312\minimum spanning t
ree using 3 methods.py
Kruskal's MST weight: 19
Prim's MST weight: 19
Borůvka's MST weight: 19
>>>
```