

103. Word Wrap Problem

AIM: To arrange a sequence of words that minimize the total cost of wrapping lines

PROGRAM:

```
def word_wrap(words, max_width):
    n = len(words)
    cost = [[0] * n for _ in range(n)]
    for i in range(n):
        cost[i][i] = max_width - len(words[i])
        for j in range(i + 1, n):
            cost[i][j] = cost[i][j - 1] - len(words[j]) - 1

    min_cost = [float('inf')] * n
    result = [0] * n
    for j in range(n):
        min_cost[j] = float('inf')
        for i in range(j + 1):
            if cost[i][j] < 0:
                continue
            if i == 0:
                current_cost = cost[i][j]
            else:
                current_cost = min_cost[i - 1] + cost[i][j] ** 2

            if current_cost < min_cost[j]:
                min_cost[j] = current_cost
                result[j] = i

    lines = []
    j = n - 1
    while j >= 0:
        i = result[j]
        lines.append(' '.join(words[i:j + 1]))
```

```
j = i - 1
```

```
lines.reverse()
```

```
return lines
```

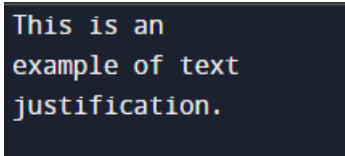
```
words = ["This", "is", "an", "example", "of", "text", "justification."]
```

```
max_width = 16
```

```
justified_lines = word_wrap(words, max_width)
```

```
for line in justified_lines:
```

```
    print(line)
```



```
This is an  
example of text  
justification.
```

OUTPUT:

TIME COMPLEXITY: $O(n^2)$