Problem Solving using Lists and Arrays
Reversing

• Rearrange elements in an array/list in the reverse order without using an additional array/list.

• L.reverse() not to be used
def reverse(L):
    n = len(L)
    for i in range(n//2):
        t = L[i]
        L[i] = L[n-1-i]
        L[n-1-i] = t
    return L
Histogram

- To find histogram capturing frequency of occurrence of an event.
  - For example, given marks for \( n \) students in the range \((0,10)\), find the count of the number of students that obtained each possible mark.
def histogram(L, size):
    hist = [0 for i in range(size+1)]
    for i in range(len(L)):
        index = L[i]
        hist[index] += 1
    return hist

def histogram2(L, size):
    hist = [0 for i in range(size+1)]
    for e in L:
        hist[e] += 1
    return hist
Largest Element

- Given a sequence (unordered) of data find the largest element

```python
def largest(L):
    largest = L[0]
    for i in range(len(L)):
        if L[i] > largest:
            largest = L[i]
    return largest
```
Search Element

- Given a sequence (unordered) of data find if an element exists

```python
def lsearch(L, x):
    for i in range(len(L)):
        if (L[i] == x):
            return i
    return -1
```
Removal of Duplicates

- Remove all duplicates from an array.
- No additional array/list to be used.
Removal of Duplicates

- Remove all duplicates from an array.
- No additional array/list to be used.

\[ \begin{array}{cccccccc}
2 & 8 & 15 & 23 & 23 & 23 & 23 & 26 \\
\end{array} \]

\[ \begin{array}{cc}
\uparrow & \uparrow \\
j & i \\
\end{array} \]

j captures unique numbers encountered
i captures number of pairs examined
Removal of Duplicates

• Remove all duplicates from an array.
• No additional array/list to be used.
Removal of Duplicates

- Remove all duplicates from an array.
- No additional array/list to be used.

```
2 2 8 15 23 23 23 26 29 30 32 32
```

```
j  i
```
Removal of Duplicates

- Remove all duplicates from an array.
- No additional array/list to be used.

A = [2, 2, 8, 15, 23, 23, 23, 23, 26, 29, 30, 32, 32]

- j
- i
Removal of Duplicates

- Remove all duplicates from an array.
- No additional array/list to be used.

```plaintext
A = [2, 8, 8, 15, 23, 23, 23, 26, 29, 30, 32, 32]
```

Removal of Duplicates

- Remove all duplicates from an array.
- No additional array/list to be used.

**Example Array A:**

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
<td>8</td>
<td>8</td>
<td>15</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>26</td>
<td>29</td>
<td>30</td>
<td>32</td>
<td>32</td>
</tr>
</tbody>
</table>

Indices: i, j
Removal of Duplicates

- Remove all duplicates from an array.
- No additional array/list to be used.

```python
A = [2, 8, 8, 15, 23, 23, 23, 23, 26, 29, 30, 32, 32]
```

```python
# Initialize j and i
j, i = 0, 1

# Loop through the array
while i < len(A):
    if A[i] != A[j]:
        j += 1
    i += 1

# Print the result
print(A)
```

Output:
```
[2, 8, 15, 23, 23, 26, 29, 30, 32, 32]
```
Removal of Duplicates

• Remove all duplicates from an array.
• No additional array/list to be used.

A = [2, 8, 15, 15, 23, 23, 23, 23, 26, 29, 30, 32, 32]

Removal of Duplicates

- Remove all duplicates from an array.
- No additional array/list to be used.

```
A = [2, 8, 15, 23, 26, 29, 30, 32, 26, 29, 30, 32, 32]
```
def remove_duplicates(L):
    n = len(L) - 1
    i, j = 0, 0
    while (i < n):
        i += 1
        if (L[i-1] != L[i]):
            j += 1
            L[j] = L[i]
    return j + 1