

Strategy: Space and Time Trade-offs (input enhancement)

Algorithm: Distribution Counting Sort [7.1]

Code:

```
def distribution_counting_sort(A, l, u):
    n = len(A)
    D = [0] * (u - l + 1)

    # Initialize frequencies
    for j in range(u - l + 1):
        D[j] = 0

    # Compute frequencies
    for i in range(n):
        D[A[i] - l] += 1

    # Reuse for distribution
    for j in range(1, u - l + 1):
        D[j] += D[j - 1]

    # Sort the array
    S = [0] * n
    for i in range(n - 1, -1, -1):
        j = A[i] - l
        S[D[j] - 1] = A[i]
        D[j] -= 1

    return S
```

Analysis:

$$\begin{aligned} &= \sum_{i=0}^{k-1} 1 + \sum_{i=0}^{n-1} 1 + \sum_{i=0}^{k-1} 1 + \sum_{i=0}^{n-1} 1 \\ &= [((k-1)-0)+1] + [(n-1)-0)+1] + [((k-1)-0)+1] + [(n-1)-0)+1] \\ &= k + n + k + n \\ &= 2n + 2k \\ &= \in \theta(n + k) \end{aligned}$$