Counting sort

```
def countingSort(A):
    # assume elements are in range 1..n
    counts = [0] * n
    for i in range(len(A)):
        counts[A[i]] += 1
    output = [ ]
    for i in range(len(counts)):
        while counts[i] > 0:
            counts[i] -= 1
            output.append(i)
    return output
```

- This takes O(n) time - independent O(n) loops
- Takes advantage of constraints on data and efficiency of modifying array elements to achieve O(1) time for each element of A
- Not comparison-based: can be shown that all comparison-based sorts take O(n log n) time

6. Python's built-in data structures