Counting Sort O(n+k)

Best sorting algorithm is $O(n\sqrt{loglogn})$ (randomized)

Assume keys are in {0,1, ..., k-1}

L = array of k empty lists

for j in 0, ..., n-1: O(n)

    L[A[j].key].append(A[j])

output = []

for i in 0, ..., k-1: O(n+k)

    output.extend(L[i])

Runtime: $\Theta(n+k)$

If $k = O(n)$, then runtime is $\Theta(n)$, beats the $\Theta(nlogn)$ lower bound since not a comparison-based sort.

As soon as k is a little bit bigger, we are in trouble.

The algorithm is stable. (Exercise)