Counting Sort

Counting sort is a linear time sorting algorithm.

Code

```ts
function countingSort(A: number[], max: number) {
    const C = new Array(max), B = new Array(A.length);
    for (let i = 0; i < C.length; i++) C[i] = 0;
    // count occurrences of each value
    for (let i = 0; i < A.length; i++) C[A[i]]++;
    // accumulate
    for (let i = 1; i < max; i++) C[i] = C[i] + C[i - 1];
    for (let i = A.length - 1; i > 0; i--) {
        B[C[A[i]]] = A[i];
        C[A[i]]--;
    }
    return B;
}
```

Design

- counting sort assumes that each of the n input elements is an integer in the range $[0, k]$ for some integer k
    - when $k = O(n)$, the sort runs in $T(n)$
    - best if $K << n$
- for each input element x, count how many elements are less than x
    - this information can be used to place x directly into its position in the output array
- does not sort in place
    - needs a 2nd array of size k and and 3rd array of size n
- it is stable

Stable Algorithms

Numbers with the *same* value appear in the output array in the same order as they do in the input array.

Stability of some other sorting algorithms

- insertion sort ✅
- quicksort ❌

Runtime Analysis

$$\Theta(n + k)$$

Where k is `max` and when $k = O(n)$ in most cases then the entire complexity is $\Theta(n)$.