8.2 Counting Sort

- Sorts an array of integers where each element is in a fixed range: $0 \rightarrow k$.

```
Counting-Sort (A, B, k)
let C[0,..,k] be new array
for i = 0 to k                          || Θ(k)
    C[i] = 0
for j = 1 to A.length                   || Θ(n)
    C[A[j]] = C[A[j]] + 1
```

```
for i = 1 to k                                  || Θ(k)
    C[i] = C[i] + C[i-1]
for j = A.length downto 1                       || Θ(n)
    B[C[A[j]]] = A[j]
    C[A[j]] = C[A[j]] - 1
```

Example: $A = [4, 1, 0, 1, 2, 4, 2, 1, 3, 2]$

$k = 4$

Create $C = \underset{0\ 1\ 2\ 3\ 4}{[0, 0, 0, 0, 0]}$

↳ $C = [1, 3, 3, 1, 2]$

↳ $C = [1, 4, 7, 8, 10]$

Note: $C[i]$ is 'the number of entries in $A' \leq i$

$\underset{1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10}{B = [\ ,\ ,\ ,\ ,\ ,\ ,\ 2,\ 3,\ ,\ ]}$

$B[C[A[10]]] = A[10]$

$B[C[2]] = 2$

$B[7] = 2$

$C[A[10]] = C[A[10]] - 1$

$C[2] = C[2] - 1$

$C = [1, 4, 6, 8, 10]$ (indices 0, 1, 2, 3, 4)

$B[C[A[9]]] = A[9]$

$B[C[3]] = 3$

$B[8] = 3$

$C[A[9]] = C[A[9]] - 1$

$C[3] = C[3] - 1$

$C = [1, 4, 6, 7, 10]$ (indices 0, 1, 2, 3, 4)

Running time of Counting Sort is

$$\Theta(k+n)$$

Thus, if $k = O(n)$

then Counting Sort is $\Theta(n)$

i.e. linear time.

[consider what Counting Sort does to

$A = [7, 9, 3, 1000, 84, 721]$.]

Note: Counting Sort is not a comparison sort.

Note: Counting Sort is stable