# Counting Sort

קלט: $n$ מספרים שלמים בטווח $[1..k]$.

נשתמש במערך עזר $C\,[1..k]$

## האלגוריתם:

$A$ - קלט. $B$ - פלט

Counting sort(A,B,k).

```
1. for i←1 to k
    1.1 C[i]←0
2. for j←1 to length(A)
    2.1 C[A[j]]←C[A[j]]+1
3. for i←2 to k
    3.1 C[i]←C[i]+C[i-1]
4. for j←length(A) down to 1
    4.1 B[C[A[j]]]←A[j]
    4.2 C[A[j]]←C[A[j]]-1
```

האלגוריתם עובד על ידי כך שאחרי שהוא סופר כמה איברים יש מכל איבר, הוא עובר על המערך המקורי ולפי הטבלה יודע איפה למקם אותו.

יש דרך יותר פשוטה - לעבור על $C$ ולשים $x$ איברים מכל ערך אחד אחרי השני. הסיבה שהאלגוריתם הזה לא עושה את זה זה כדי להיות מיון יציב:

#### מיון יציב(Stable)

מספרים עם אותו ערך יופיעו במערך הפלט באותו סדר שבו הם הופיעו במערך הקלט.

#### סיבוכיות

חלק 1 לוקח $\theta\,(k)$. חלק 2 לוקח $\theta\,(n)$. חלק 3 לוקח $\theta\,(k)$. חלק 4 לוקח $\theta\,(n)$
זמן ריצה - $\theta\,(n+k)$
אם $k=O\,(n)$ אז המיון בזמן $\theta\,(n)$.

# מיון בסיס Radix sort

קלט: $n$ מספרים בעלי $d$ ספרות בטווח $[0..k-1]$(כלומר כל ספרה היא בבסיס $k$)

הרעיון: נעשה מיונים לפי ספרות, כשמתחילים מהספרה הפחות משמעותית $LSD$.

## האלגוריתם

$A$ - קלט

<u>Radix-sort(A,d)</u>
```
1. for i←1 to d

    1.1 sort array A on digit i using stable sort
```

### סיבוכיות

מבצעים Counting sort $d$ פעמים. Counting sort דורש $\theta\left(n+k\right)$ זמן. סה"כ זמן ריצה אם משתמשים בCounting sort - $\theta\left(d\left(n+k\right)\right)$. אם $d$ קבוע ו$k=O\left(n\right)$, זמן הריצה הוא $\theta\left(n\right)$.

## טענה

Radix sort ממיין נכון והוא מיון יציב.

### הוכחה

באינדוקציה על מספר הספרות.
נגדיר $x_l$- $l$ הספרות הפחות משמעותיות של המס' $x$.

<u>בסיס</u> $d=1$: ביצענו מיון יציב על $n$ מספרים בטווח $[0,...k-1]$ ולכן אם $x_1<y_1$ אז $x$ יופיע לפני $y$, ואם $x_1=y_1$ המיקום היחסי לא ישתנה כי המיון יציב.

<u>הנחת האינדוקציה</u> $d=l-1$: אם $x_{l-1}<y_{l-1}$ אז $x$ יופיע לפני $y$, ואם $x_{l-1}=y_{l-1}$ אז המיקום היחסי לא השתנה.

<u>צעד:</u> $d=l$: אם $x_l<y_l$ אז יש שתי אפשרויות:

1) הספרה $l$ של $x$ קטנה מהספרה ה$l$ של $y$. המיון לפי הספרה ה$l$ ישים את $x$ לפני $y$.

2) הספרה ה$l$ שווה ב$x$ וב$y$, ומתקיים $x_{l-1}<x_{l+1}$. לפי הנחת האינדוקציה, לפני המיון של הספרה ה$l$, $x$ הופיע לפני $y$. מכיוון שהספרה ה$l$ זהה והמיון יציב, $x$ ישאר לפני $y$.

אם $x_l=y_l$ אז $x_{l-1}=y_{l-1}$ אז לפי הנחת האינדוקציה המיקום היחסי <u>לפני</u> המיון של הספרה $l$, נשמר. ומכיוון שהמיון לפי הספרה ה$l$ יציב נקבל שהמיקום היחסי נשמר.