# Linear time sorting

CS 146 - Spring 2017

# Today

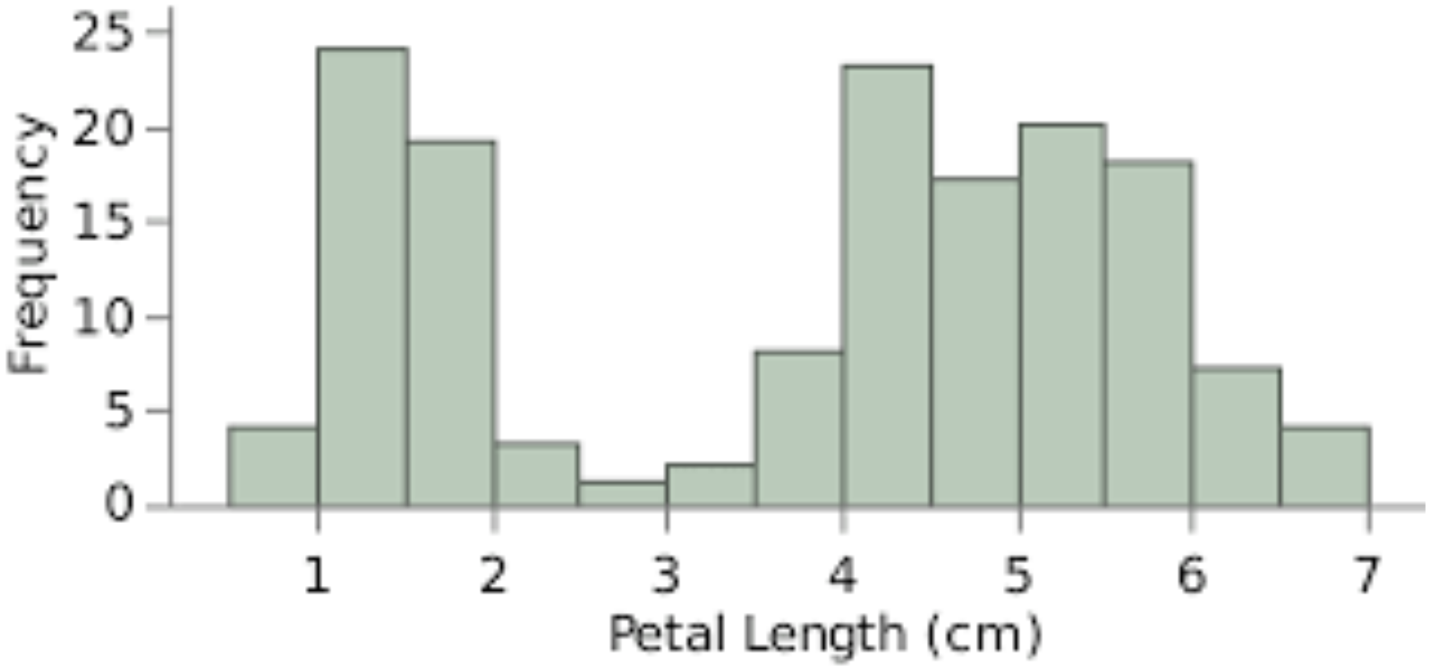
- counting sort
- bucket sort
- radix sort

# Last time…
# How would you sort…

1 trillion integers, all of which are between 1 and 10?

# Counting sort

idea
build a
histogram



**input**: list of integers, **upper bound on these ints**

**output**: list of integers, sorted

Time: **$\Theta(n+k)$**, $n$ = number of ints, $k$ = upper bound

https://en.wikipedia.org/wiki/Histogram

# Bucket sort

**input**: list of integers

**a function by which to sort by**

**upper bound on the range of the function**

**output**: list of integers, sorted according to the function

number of buckets

Time: **$\Theta(n+B)$**, n = number of ints, B = upper bound of sort-by function

# Stable sorting algorithm

a sorting algorithm which preserves the input order among equal elements.

Bucket sort is stable. Why?

# Card sort

https://youtu.be/VQueCt114Gk?t=261

# Radix sort

**input**: list of integers, **upper bound on these ints**

**algorithm parameter** : radix or base b

**output**: list of integers, sorted

Time: **ϴ(dn)**, n = number of ints, input values $\leq n^d$