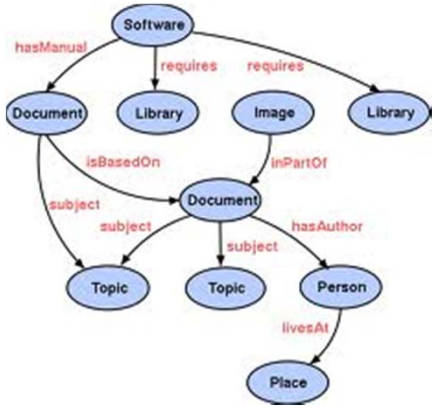
What apps naturally deal w/ graphs?

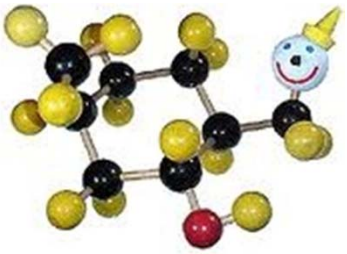
Social Networks

Semantic Web

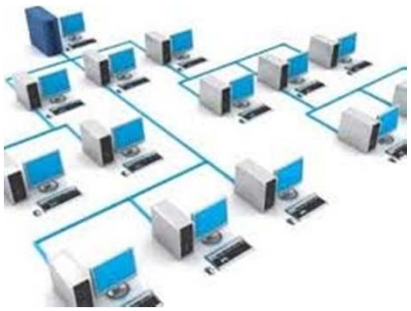


World Wide Web

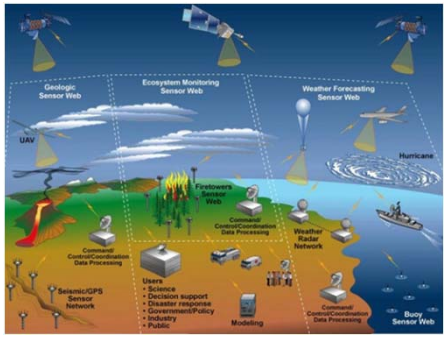
Drug Design,
Chemical compounds



Computer networks



Sensor networks



Credit: Images are from Google images via search of keywords

How Big Are These Graphs?

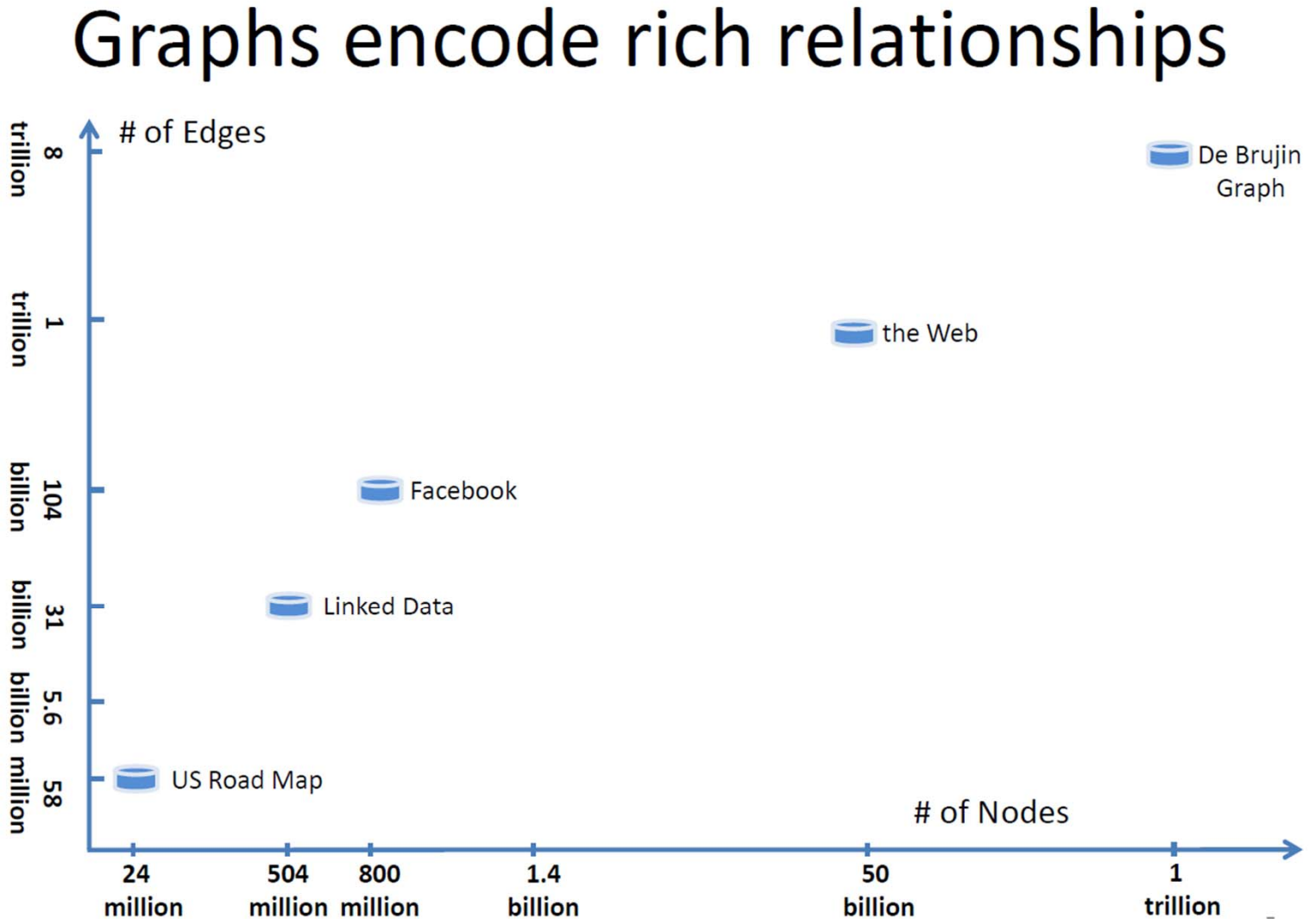


Image source: Haixun Wang, KDD 2012

What Algorithms to Use to Analyze Graphs?

How practical are these algorithm, in terms of Big-O?

Analysis algorithms fail for a few gigabytes.

Algorithmic Complexity:

Calculate means	O(*n*)
Sorting	O(*n* log(*n*))
Calculate SVD	O(*r* • *c*)
Clustering algorithms	O(n^2)

If *n*=10GB, then what is *O*(*n*) or *O*(n^2) on a teraflop computers?

1GB = 10^9 bytes
1Tflop = 10^{12} op/sec

Data size *n*	Algorithm Complexity		
	n	***nlog(n)***	n^2
100B	10^{-10} sec.	10^{-10} sec.	10^{-8} sec.
10KB	10^{-8} sec.	10^{-8} sec.	10^{-4} sec.
1MB	10^{-6} sec.	10^{-5} sec.	1 sec.
100MB	10^{-4} sec.	10^{-3} sec.	3 hrs
10GB	10^{-2} sec.	0.1 sec.	**3 yrs.**

For illustration chart assumes 10^{-12} sec. (**1Tflop/sec**) calculation time per data point

Quick refresher on Big O notation

An algorithm has time complexity ***O(f(n))*** if there is some constant ***c*** such that the algorithm runs for no more than ***c•f(n)*** steps on an input of size ***n***.

- Worst case complexity
- Upper bound
- Don't worry about constant multiples

  Ex) "Algorithm A runs twice as fast as Algorithm B."

Problem	Complexity
Sorting	$O(n \log n)$ (Quicksort, Mergesort, et al.)
Sum	$O(n)$
Search	$O(n)$ (unsorted)
	$O(\log n)$ (sorted)

Example: Steps to Evaluate Big-O

- **Let f(x)=x^2+3x+1000**
- **To evaluate O(f(x)):**
  1. For all sums, replace all terms in a sum with the largest.
  2. Add/multiply remaining terms.
  3. Drop constants.