Tractability

What Computers Can and Cannot Do

Back to School ...

n	$x=10^n$
0	1
1	10
2	100
3	1,000
4	10,000
5	100,000
6	1,000,000
7	10,000,000
8	100,000,000
9	1,000,000,000
10	10,000,000,000

x	$n=\log_{10}x$
1	0
10	1
100	2
1,000	3
10,000	4
100,000	5
1,000,000	6
10,000,000	7
100,000,000	8
1,000,000,000	9
10,000,000,000	10

By definition, $\log_{10}(10^n)=n$

… or in Base 2 …

n	$x=2^n$
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1,024

x	$n=\log_2 x$
1	0
2	1
4	2
8	3
16	4
32	5
64	6
128	7
256	8
512	9
1,024	10

By definition, $\log_2(2^n)=n$

Some Simple Problems

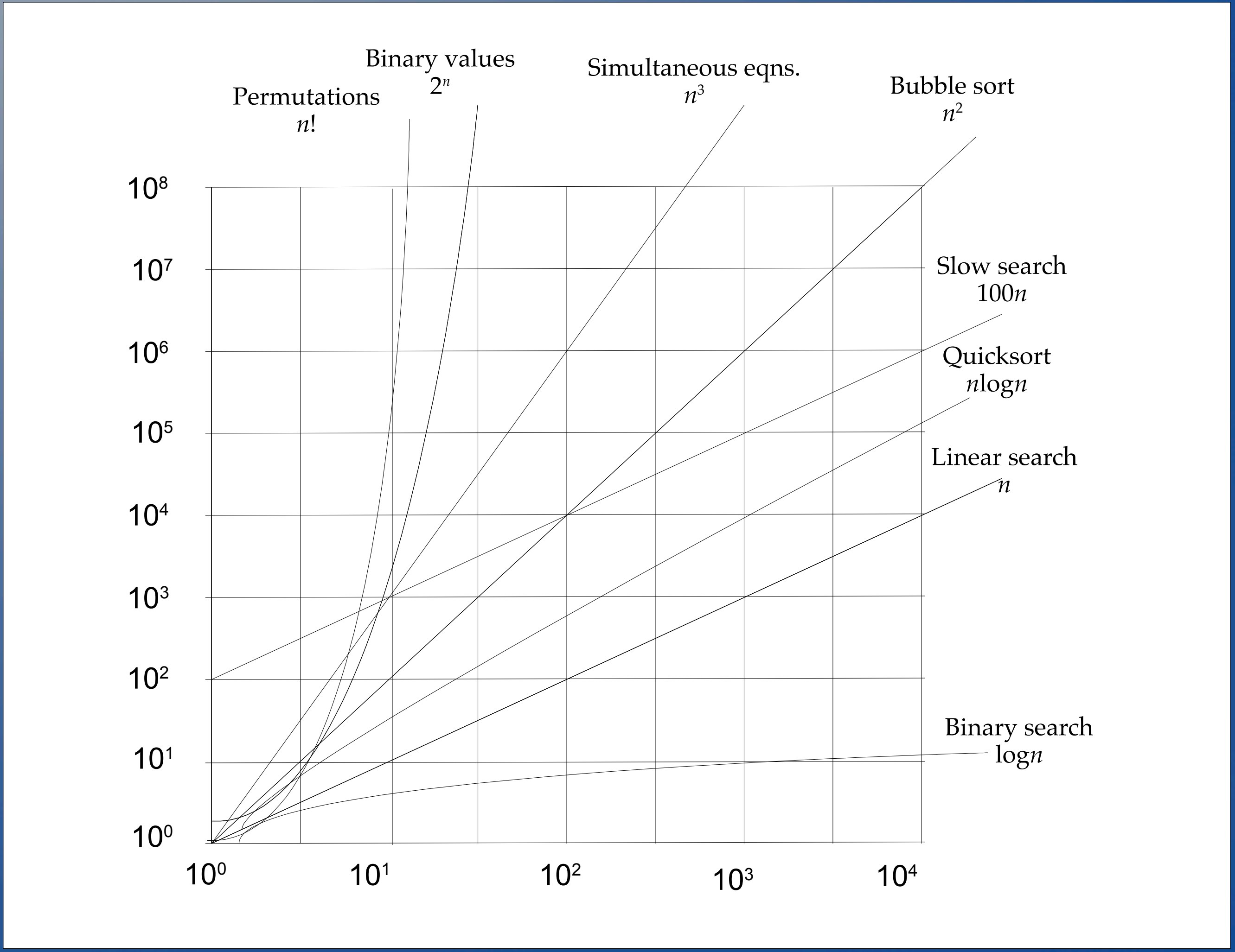
- Find a card in a shuffled deck of *n* cards
  - Time proportional to n; $t = \mathbf{O}(n)$
- Find a page in a phone book of *n* pages
  - Time proportional to $\log_2 n$; $t = \mathbf{O}(\log_2 n)$
- Sort a deck of *n* cards using bubble-sort, etc.
  - Time proportional to n^2; $t = \mathbf{O}(n^2)$
- Sort a deck of cards using merge-sort, etc.
  - Time proportional to $n\log_2 n$; $t = \mathbf{O}(n\log_2 n)$

Some Hard Problems

- Listing all binary numbers of length n:
  - $\mathbf{O}(2^n)$
- Listing all permutations of a sequence of length n:
  - $\mathbf{O}(n \times (n-1) \times (n-2) \times (n-3) \ldots \times 2 \times 1) = \mathbf{O}(n!)$
- Scheduling exam time-table in the fewest sessions:
  - maps onto Graph Colouring problem
- Travelling salesman problem:
  - a version of Hamiltonian Circuit problem

Comparative Growths

- On log-log scale kn^x is a straight line

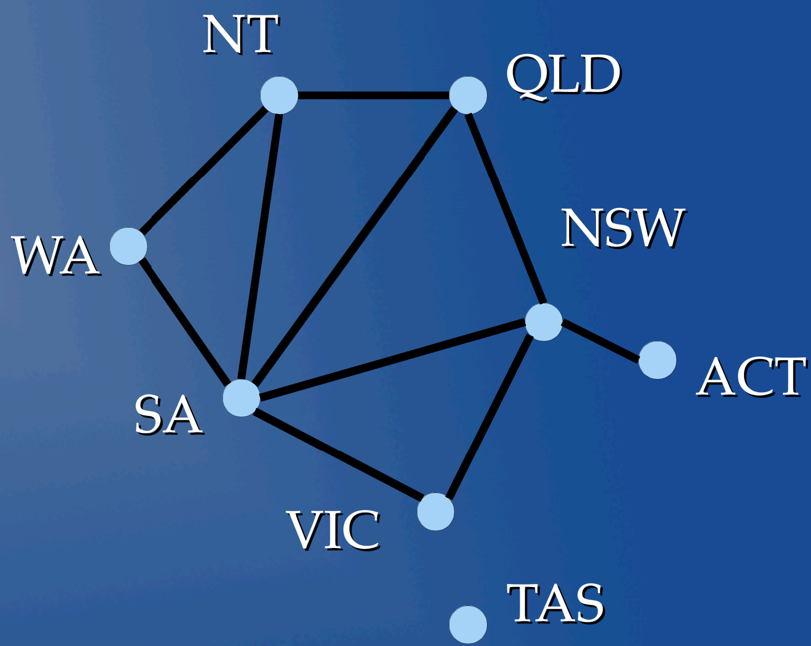


Definition of Tractability

- A problem is *tractable* if an algorithm exists for its solution that takes time that is bounded by a *finite* polynomial in the size of the input
- In other words, its graph never rises above some straight line.
- All other problems are *intractable*.

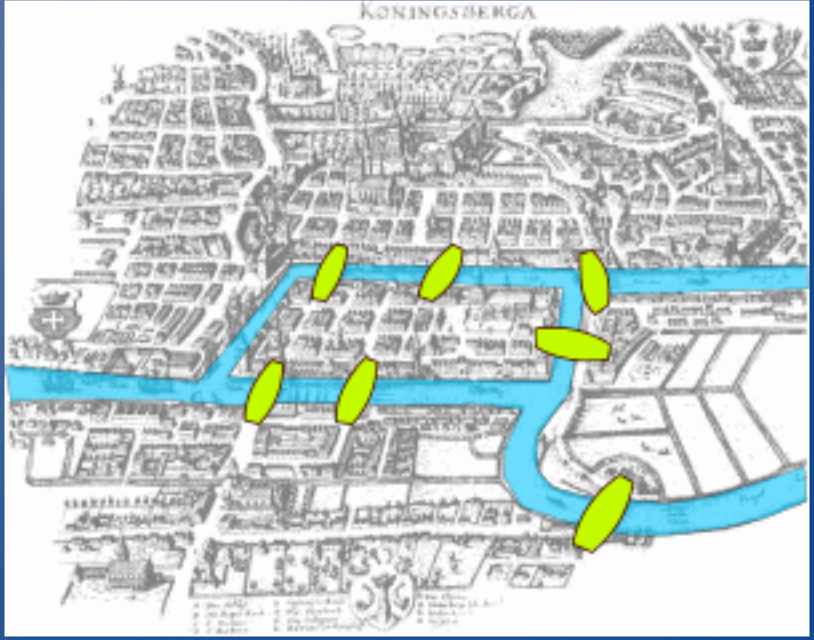
Graph Theory can Describe Problems

- Graphs can be used to remove needless detail:
  - Points are called vertices
  - Lines are called edges



Hard or Easy?

- Euler Circuit: cross every bridge — easy
- Hamiltonian Circuit: visit every place — hard
- Find all anagrams of a given word:
    - Permute letters: **O**(n!)
    - Scan dictionary: **O**(1)



Degrees of Difficulty

- Insoluble
  - Halting problem, paradoxes.
- Intractable
  - All permutations
- NP-complete
  - Read on ...
- Maybe NP-hard
  - graph isomorphism
- Polynomial

3-Sat: The 1st NP-complete Problem

Find values of A, B, C, and D that make
(A or D) & (B or $\overline{C}$) & (C or $\overline{A}$) & (A or $\overline{B}$ or $\overline{C}$) & ($\overline{B}$ or $\overline{C}$)
true.

(Easy to make it false, e.g., A false, D false.)

	$\overline{A}$ $\overline{B}$ $\overline{C}$ $\overline{D}$	$\overline{A}$ $\overline{B}$ $\overline{C}$ D	$\overline{A}$ $\overline{B}$ C $\overline{D}$	$\overline{A}$ $\overline{B}$ C D	$\overline{A}$ B $\overline{C}$ $\overline{D}$	$\overline{A}$ B $\overline{C}$ D	$\overline{A}$ B C $\overline{D}$	$\overline{A}$ B C D	A $\overline{B}$ $\overline{C}$ $\overline{D}$	A $\overline{B}$ $\overline{C}$ D	A $\overline{B}$ C $\overline{D}$	A $\overline{B}$ C D	A B $\overline{C}$ $\overline{D}$	A B $\overline{C}$ D	A B C $\overline{D}$	A B C D
A or D		T		T		T		T	T	T	T	T	T	T	T	T
B or $\overline{C}$	T	T			T	T	T	T	T	T			T	T	T	T
C or $\overline{A}$	T	T	T	T	T	T	T	T			T	T			T	T
A or $\overline{B}$ or $\overline{C}$	T	T	T	T	T	T			T	T	T	T	T	T	T	T
$\overline{B}$ or $\overline{C}$	T	T	T	T	T	T			T	T	T	T	T	T		

Knowing you have an NP problem

- If solving your problem would solve a known NP problem, it is NP-hard.
- If solving a known NP-complete problem would solve your NP-hard problem, it is NP-complete.
- Exam timetable:
    - Use the fewest sessions.
    - Avoid timetable clashes.
    - Two subjects clash if at least one student sits both.
    - Heuristic: biggest classes first.

Graph Isomorphism—P or NP?

- Are any two graphs the same after relabelling?