Classical and Quantum Complexity Classes

ECE 592-100
March 2023

Dror Baron
Associate Professor
Dept. of Electrical and Computer Engr.
North Carolina State University, NC, USA

Two sorting algorithms

- Want to sort n real-valued numbers

- Insert sort
  - Maintain (sorted) list of numbers processed so far
  - Next item gets inserted into list

- Merge sort
  - Divide problem into two parts (roughly equal size)
  - Conquer each problem (run merge sort recursively)
  - Merge solutions

How many operations do they perform?

- Denote # ops for sorting n numbers by T(n)

- Insert sort: $T_i(n) \propto n^2$ (proportional to)
  - Rationale: adding n'th number could take n operations; $1+2+\ldots+n = n(n+1)/2 \propto n^2$

- Merge sort: $T_m(n) \propto n \times \log_2(n)$
  - Rationale: $T_m(n) = 2 \times T_m(n/2) + n$ (merging takes n)
  - Recursive formula; can show it's $\propto n \times \log_2(n)$

Example running times

- Give insert sort an edge
  - Merge implemented by bad programmer → $100n \times \log_2(n)$
  - Insert runs on cluster (10^{12} operations/sec)
  - Merge runs on regular machine (10^9)
- $n=10^3$: Merge sort 1 ms, Insert sort 1 us
- $n=10^6$: Merge 2 s, Insert 1 s
- $n=10^9$: Merge 3000 s (50 minutes), Insert 11 days
- *Asymptotic growth matters*

Computational complexity

- Consider $T(n)=an^2+bn+c$
  - a, b, c positive constants
- Asymptotically, an^2 matters
  - bn+c doesn't
- Need to characterize asymptotic growth → computational complexity
  - Complexity of T(n) is n^2; grows like n^2

Complexity Classes

Classical complexity classes

- Main distinction *polynomial* vs. *exponential* runtime
- For exp. algos, multiplicative speed increase provides additive problem size increase

Classical complexity classes

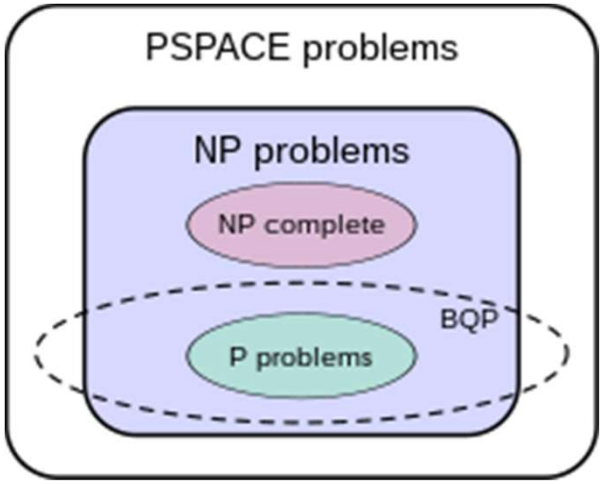
- Main distinction *polynomial* vs. *exponential* runtime
- For exp. algos, multiplicative speed increase provides additive problem size increase

- P – class of problems with poly runtime
- NP – class whose solution can be verified w/ poly runtime
- PSPACE – class w/ poly memory use
- EXP – class w/ exp runtime

Quantum complexity class BQP

- Bounded probability of error (B)
  - You have 10% error? ☹
  - No problem! Run it k times ☺
- Quantum algorithm (Q)
- Polynomial number of quantum gates (P)

BQP versus classical



- $P \subseteq BQP$
  - Poly quantum gates emulate poly classical
- $BQP \subseteq PSPACE$
  - Poly quantum gates can be simulated w/ poly classical memory
- BQP seems to contain problems beyond NP
  - Deutsch-Jozsa & Simon's algorithms replace exponential classical queries by 1 & n quantum queries, respectively