Data checking and verification Randomization and coding

A guide to PCP, an approach towards classification of NP problems

P versus NP

Have you heard about P vs NP problem?

- Millenium prize problems by Cray institute
  http://www.claymath.org/millennium/

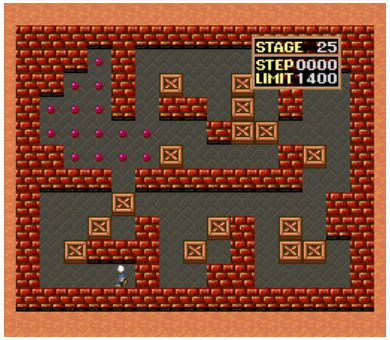
- P versus NP is not only million dollar problem, Its value is trillion dollars.

Class P and class NP

- Consider a computational problem M of input size n
- P: class of polynomial time solvable problems in usual model of computation
  - M is in P if there is an algorithm to solve it in polynomial time
    - Example: Sorting, Searching, Graph connectivity
    - Shortest path, text pattern matching
- NP: class of polynomial time solvable problems in nondeterministic model of computation
  - M is in NP if there is a nondeterministic algorithm to solve it in polynomial time (difficult to understand, isn't it?)
- P is a subclass of NP

More intuitive description

- Verification is often easier than solution
  - Solving an equation is more difficult than verification
- NP: If we know the solution, we can show it such that verification is done in polynomial time
  - 1: Hamilton path
  - 2: Graph coloring,
  - 3: Does this SUDOK puzzle have a solution?
- A problem (probably) not in NP
  - 1: Will black win this Chess game ?
  - 2: SOKOBAN game
  - 3: #3SAT (number 3SAT)



- Intuitively, a problem that we are convinced with correctness of solution easily if it is shown.

NP and co-NP

- It is notable that negation of an NP problem might not be in NP
  - Is a given graph G is colored by using k colors?
    - We give the actual coloring to veryfy. Thus NP problem
  - Is a given graph G has no coloring with k colors?
    - If so, how to convince it?
- Co-NP problem: negation of an NP problem
- It is believed NP and co-NP are different
- It might be true that NP ∩ coNP = P

P vs NP

- Is it true that verification is easier than solution ?
  - Consider an equation $f(x) = x^5 - 3x^4 + 3x^2 + 4 = 0$
  - Ask " is this equation has an integer solution less than n (say, n=100)".
  - Verification: f(2) = 32-48+12+4 = 0
  - Solution???
- P vs NP: Is there any problem where verification is easy (P-time) but solution is difficult (not in P-time)?
- Philosophical question
  - Is it easy to learn than solve by yourself?
    - Is it true that a difficult problem remains to be difficult even if you are suggested a solution (you must make sure it is true).
  - In real life, we must solve many NP problems(or even more difficult problems). Human can solve them by training (like SUDOK) for most of instances. Why??

Implication of P=NP

- If P=NP, we can solve many problems
  - Many-body problem in physics
  - Protein folding problem in biology
  - Optimal scheduling in manufacturing
  - Optimal traffic control
  - Many problems in computational chemistry
- If P=NP, we have serious inconvenience
  - Current cryptology assumes P is not NP
  - Information security system is destroyed
- Most of researchers believes P is not NP, and the above situation only occurs in Scientific Fiction
  - But no one knows the truth
- The biggest mathematical challenge in 21st century

Several approaches towards P vs NP

http://people.cs.uchicago.edu/~fortnow/papers/pnp-cacm.pdf

Lance Fortnow's article

- NP- complete theory (S.Cook and R.Karp)
  - Almost all NP problems are either in P or NP-complete(one of the most difficult problems in NP)
- Circuit complexity
  - Show NP needs exponential size of circuits
- Algebraic/group theoretic method
  - Relation to generalized Riemann hypothesis (Mulmuley)
- From mathematical logic
- Relation to randomness
- **Interactive Proof and Checkable Proof**

Textbooks/papers

- Randomized Algorithms (Motwani-Raghavan)
- Efficient Checking of Polynomials and Proofs and the Hardness of Approximation Problems

  (M. Sudan, ACM Distinguished Thesis, 1995)
- Proof Verification and the Hardness of Approximation Problems (S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy), J. ACM, Vol 45(3), 1998, pp. 501-555

Rajeev Motwani (from wikipedia)

- Motwani joined Stanford soon after U.C. Berkeley. Motwani was one of the co-authors (with Larry Page and Sergey Brin, and Terry Winograd) of an influential early paper on the PageRank algorithm, the basis for Google's search techniques. He also co-authored another seminal search paper *What Can You Do With A Web In Your Pocket* with those same authors.[3]
- He was also an author of two widely-used theoretical computer science textbooks, *Randomized Algorithms* (Cambridge University Press 1995, ISBN 978-0521474658, with Prabhakar Raghavan) and *Introduction to Automata Theory, Languages, and Computation* (2nd ed., Addison-Wesley, 2000, with John Hopcroft and Jeffrey Ullman).
- Prior to his involvement with Google, Motwani founded the Mining Data at Stanford project (MIDAS), an umbrella organization for several groups looking into new and innovative data management concepts. His research included data privacy, web search, robotics, and computational drug design.
- He was an avid angel investor and had funded a number of successful startups to emerge from Stanford. He sat on the boards of Google, Kaboodle, Mimosa Systems, Adchemy, Baynote, Vuclip, NeoPath Networks (acquired by Cisco Systems in 2007), Tapulous and Stanford Student Enterprises among others. He was also active in the Business Association of Stanford Entrepeneurial Students (BASES).[4][5][6]
- He was a winner of the Gödel Prize in 2001 for his work on the PCP theorem and its applications to hardness of approximation.[7][8]

Verification

Algebraic methods to verify information

Verification in real life

- Given two data A and B, we want to verify A=B
  - We want to do it without reading A and B explicitly
- If A and B are “persons”
  - How we can “read” data of persons completely?
    - Impossible!
  - Name, blood type, color of hair/eye,……
  - ID numbers, secret keywords,
  - Finger print, DNA identification
    - If A and B are twins……..

Fingerprinting

- We recall the data checking problem we discussed some weeks ago
  - Given a set of n data {a(1),a(2),…,a(n)}, and we suspect one (or more) data is modified.
  - How should we check efficiently?
    - The sum of all data
    - Use hash function, and sum of h(a(i))
  - This is an example of “fingerprint”

Fingerprinting for matrix multiplication

- Consider a prime p
- Given n by n matrices A, B, and C, we want to verify AB = C mod p
  - Computation in the field GF(p) (or $\mathbf{Z_p}$)
- Can you do it in $O(n^2)$ time?

Verifying an identity

- $X = (x_1, x_2, .., x_d)$
- We want to verify a polynomial identity $F_1(X) = F_2(X)$ of degree n

$$\Delta(x_1, \ldots, x_n) \equiv \begin{vmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{vmatrix}$$

$$= \prod_{i,j} (x_i - x_j)$$

Verifying an identity

- $X = (x_1, x_2, .., x_d)$
- We want to verify a polynomial identity $F_1(X)=F_2(X)$ of degree n
  - Or more cruel identities

THEOREM 5.1 (Tokuyama [70]) *We have*

$$\sum_{\substack{\mathfrak{T}\in \mathrm{GT}(\lambda+\rho) \\ \mathfrak{T} \text{ strict}}} (t+1)^{s(\mathfrak{T})} t^{l(\mathfrak{T})} z_1^{d_r} z_2^{d_{r-1}-d_r} \cdots z_{r+1}^{d_0-d_1} =$$

$$\left\{ \prod_{i<j} (z_j + z_i t) \right\} s_\lambda(z_1, \cdots, z_{r+1}).$$

Verifying strings (communication complexity)

- Alice and Bob have very long documents A and B (of length n each).
- They want to confirm they are identical.
  - They speak on wire, and cannot tell the whole document.
- How is your solution?
  - Assume that they have a program to generate a large prime number.

Pattern matching via verification

- Given a text T of length n (bit string)
- For each query pattern P of length m, we want to find location of occurrence of P in T
- Both m and n are long (say, m = 100000, n = 10000000)
- KMP algorithm, BM algorithm: optimal O(n+m), but not much practical
- Can we apply verification idea?