[Chapter 9]

Computational Complexity and Intractability: An Introduction to the Theory of NP

Scenario from Garey and Johnson (1979)

- Task of finding an efficient algorithm for Problem X.
- No headway at all toward an efficient algorithm.
- Your boss threatens to fire you.
  - You reply that perhaps it is not because you're stupid but it is not possible.
  - Your boss gives you another month to prove your claim.
- No proof that it is not possible.
  - On the verge of being fired, you come to think of the fact that nobody developed an efficient algorithm for TSP.
  - Now you see one last glimmer of hope.
  - Can you prove that Algorithm X can produce an efficient algorithm for TSP?
- No firing, but promotion!!

Intractability

- Polynomial-time algorithm
  - $W(n) \in O(p(n))$
  - That is, if n is the input size, there exits a polynomial $p(n)$
  - (e.g.) $2n, 3n^3 + 4n, 5n + n^{10}, n \lg n$
  - (e.g.) $-\ 2^n, 2^{0.01n}, 2^{\sqrt{n}}, n!$
- Problem X is "intractable"
  - If it is impossible to solve it with a polynomial-time algorithm.
  - It is a property of a problem.
  - It is not a property of any algorithm for that problem.
  - (e.g.) chain of matrix multiplication is tractable.

Intractability (Cont'd)

- Suppose n=10^6
  - $$n^{10} = 10^{60}$$
    $$vs$$
    $$2^{0.00001n} = 2^{10} = 1{,}024$$
  - Sometimes, many algorithms whose worst-case time complexities are not polynomials have efficient running times for many practical instances.
  - (e.g.) Decision algorithm for Presburger arithmetic is proven to be $2^{2^{cn}}$ or $2^{2^{2^{cn}}}$
    - William Pugh, *The Omega test: a fast and practical integer programming algorithm for dependence analysis*, 1991.

Intractability: 3 general categories

- Problems for which polynomial-time algorithms have been found.
- Problems that have been proven to be intractable.
- Problems that have not been proven to be intractable, but for which polynomial-time algorithms have never been found.
  - Almost all meaningful problems in computer science fall into either the first or the third.

Input Size Revisited

- Prime number check
  - ```
    while ( i <= floor(sqrt(n)))
        if (n % i == 0) return (“not prime”)
    ```
- Time complexity $\Theta(n^{1/2})$?
- Hmm, I don’t think so, because n is not an input size. It is simply an input value.
- Definition
  - For a given algorithm, the input size is defined as the number of characters it takes to write the input.
  - (e.g.) a positive integer x → $\lg x + 1$
  - 31 = 11111_2
    ```

Input Size Revisited (Cont'd)

- Definition
  - For a given algorithm, $W(s)$ is defined as the maximum number of steps done by the algorithm for an input size of s. $W(s)$ is called the worst-case time complexity of the algorithm.
  - (e.g.) Prime check: time complexity analysis using $s = \lg n$ instead of n. (binary encoding)
  - $$\Theta(n^{1/2}) = \Theta(2^{s/2})$$
  - Exercise for ExchangeSort() in terms of s
    - The number of inc & branch:
    - The number of comparisons:
    - The number of assignments:

Three General Problem Categories

- Problems for which polynomial-time algorithms have been found
  - Most of problems from Chaps. 1-6
- Problems that have not been proven to be intractable but for which polynomial-time algorithms have never been found.
  - 0/1 Knapsack, TSP, m-coloring problem, Hamiltonian circuit problem, ...
  - Even though we don’t know about its lower bound yet, there is a close and interesting relationship among these problems.

Three General Problem Categories

- Problems that have been proven to be intractable
  - Case 1: impractical problem

    (e.g.) Determining all Hamiltonian circuits in a complete graph: $(n\text{-}1)!$
  - Case 2: Practical case: surprisingly very rare!
    - Undecidable problem

      (e.g.) Halting problem in a Turing machine.
    - Decidable problem
    - Decidably intractable problem by Grzegorczyk (1953)
    - Another one by Hartmanis & Sterns (1965)
    - Decision problem for Presburger Arithmetic by Fisher & Rabin in 1974 for double exponential and by Oppen in 1978 for triple exponential ones, respectively.