Complexit Classes

Note Title 9/8/2019

Recall : complexity notations

The limit rule :

let $L = \lim_{n \to \infty} \frac{f(n)}{g(n)}$

then :

1. if $L < \infty$, $f(n) = O(g(n))$
2. if $L > 0$, $f(n) = \Omega(g(n))$
3. if $L = c > 0$, $f(n) = \Theta(g(n))$
4. if $L = 0$, $f(n) = o(g(n))$

Note:

$f = O(g)$: $f \leq g$

$f = \Omega(g)$: $f \geq g$

$f = \Theta(g)$: $f \cong g$

$f = o(g)$: $f < g$

§ 9.2. Class P:

1] Def^n. Algorithm A is deterministic if it has only one choice (determined) in each step.

2] Def^n. The Class P: all decision problems that can be solved using a deterministic algorithm in polynomial time $O(n^k)$

e.g.

Sorting: Is-sorted? $\in P$

2-coloring: Is G 2-colorable? $\in P$

G is 2-colorable iff no odd cycles

3] Thrm 1: Class P is closed under complementation.

e.g. Not-2-Colorable $\in$ P

4] The Class NP: all decision problems that can be solved using a nondeterministic algorithm in poly-time.

i.e. the algorithm can "guess" a solution in $O(n^i)$

then "verify" it in $O(n^j)$

Total time: $O(n^i) + O(n^j) = O(n^k)$ poly-time.

e.g. coloring $\in$ NP

5] Prove that coloring $\in NP$

informal: if we are given a coloring assigment $c(x)$, $\forall x \in V$, then we can verify it in $O(n^2)$ by examinning each $(x,y) \in E$, $c(x) \neq c(y)$

6] P vs. NP:

$\left.\begin{array}{l}\text{P problems can be solved in poly-time} \\ \text{NP \quad " \quad can be verified in poly-time}\end{array}\right\}$ using deterministic algorithm.

§ 9.4. NP - Complete Problems

" The class of the hardest NP problems "

7] Def^n. Let A and B be 2 decision problems, then

$A \propto_p B$ means: A is reduced to B in poly-time.

" solving B implies solving A "

" A is easier than B " " $A \leq B$ "

8] Thrm 2: if $A \propto_p B$ and $B \propto_p C$ then $A \propto_p C$

i.e. " $\propto$ is transitive "