Blackboard Collab has some lectures from prev semesters.

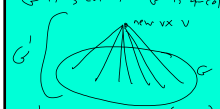
Quiz 6 — get an email with the quiz at 11am this Monday. Upload answers (pics, scan, type) to CSE Dropbox.

[No class next week]

Prop: 3-COL $\leq_p$ 4-COL

Cor: 4-COL is NP-complete.

Proof: Given a graph G, build in ptime a graph G' s.t. G is 3-col iff G' is 4-col.

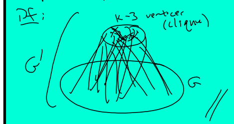


$G'.V = G.V \cup \{v\}$ $(v \notin G.V)$

$G'.E = G.E \cup \{(u,v) : u \in G.V\}$ //

Prop: $\forall k \geq 3$, 3-COL $\leq_p$ k-COL.

Pf:



//

Def: SUBSET-SUM is the following decision problem:

Instance: A list of distinct positive integers $\langle a_1, a_2, \ldots, a_n \rangle$ and a positive integer t (the "target"). [all #'s represented in binary]

Question: Is there a set $J \subseteq \{1, \ldots, n\}$ such that

$$\sum_{i \in J} a_i = t \ ?$$

Prop: SUBSET-SUM is NP-complete.

Proof: SUBSET-SUM $\in$ NP:

Easily verifiable proof of membership (yes-instance) a set J that works.

For NP-hardness:

3-SAT $\leq_p$ SUBSET-SUM.

Input: A 3-cnf fmla

$$\varphi = C_1 \wedge \cdots \wedge C_k$$

over vars $x_1, \ldots, x_n$.

For the output list of numbers:

$\langle a_{x_1}, a_{\bar{x}_1}, a_{x_2}, a_{\bar{x}_2}, \ldots, a_{x_n}, a_{\bar{x}_n}, \ldots \rangle$

For any literal ℓ, a_ℓ has the form:

$$\begin{array}{cccc|c} C_1 & C_2 & & C_k & \overbrace{\quad\quad\quad\quad}^{n \text{ bits}} \\ b_{\ell,1} & b_{\ell,2} & \cdots & b_{\ell,k} & 0 \cdots 0\,1\,0 \cdots 0 \end{array}$$

3-bit blocks

where $b_{\ell,i} = \begin{cases} 000 & \text{if } \ell \text{ does not appear in } C_i \\ 001 & \text{if } \ell \text{ does appear in } C_i \end{cases}$

last n-bits of a_ℓ are $0 - 0\,1\,0 - 0$ (jth bit) if $\ell = x_j$ or $\bar{x}_j$.

Ex: $\varphi = (\bar{x}_1 \vee x_2 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \bar{x}_3)$

$a_{x_1} = 000\,|\,000\,|\,001\,\|\,1000$

$a_{\bar{x}_1} = 001\,|\,001\,|\,000\,\|\,1000$

$a_{x_2} = 001\,|\,000\,|\,001\,\|\,0100$

$a_{\bar{x}_2} = 000\,|\,001\,|\,000\,\|\,0100$

$\qquad\qquad\qquad\qquad\qquad\quad 0010$

$\qquad\qquad\qquad\qquad\qquad\quad 0010$

$\vdots \qquad\qquad\qquad\qquad\quad\ \ 0001$

$t := \underbrace{100}_{4}\,|\,\underbrace{100}_{4}\,|\,\underbrace{100}_{4}\,\|\,1111$

[t has 100 in each 3-bit block & last n bits all 1's.]

Any set of a_ℓ adding to t must have exactly one of a_x and $a_{\bar{x}}$.

a_x: set x true; $a_{\bar{x}}$: set x false

Slack numbers: for each $1 \leq i \leq k$ include in the list

$a_{i,1}$ $0 - 0\,|\,\overset{C_i}{001}\,|\,0 - 0\,\|\,0 - 0$

$a_{i,2}$ $0 - 0\,|\,010\,|\,0 - 0\,\|\,0 - 0$

End-of-construction.

φ is sat $\iff$ t is sum of a subset of the a's //

PARTITION:

Input: A list $\langle a_1, \dots, a_n \rangle$ of positive ints in binary.

Question: Is there a set $J \subseteq \{1, \dots, n\}$ such that

$$\sum_{i \in J} a_i = \sum_{i \notin J} a_i \; ?$$

[essentially a restriction of SUBSET-SUM where $t = \frac{1}{2} \sum_{i=1}^{n} a_i$]

Prop: SUBSET-SUM $\leq_p$ PARTITION.

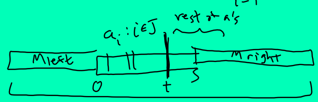
Proof Idea:

Given $\langle a_1, \dots, a_n \rangle$ and t

construct $\langle a_1, \dots, a_n, M_{left}, M_{right} \rangle$

If $t > \sum_{i=1}^{n} a_i$ then output $\langle 1 \rangle$ [no-instance]

Otherwise $0 \leq t \leq S := \sum_{i=1}^{n} a_i$



Want: $M_{left} + t = M_{right} + S - t$

First: choose M_{left} sufficiently large. Then set $M_{right} := M_{left} - S + 2t$

Choose M_{left} so that

$$M_{left} + M_{right} > \frac{S + M_{left} + M_{right}}{2}$$

[M_{left} & M_{right} can't be on the same side of a partition)

$$2M_{left} + 2M_{right} > S + M_{left} + M_{right}$$

$$M_{left} + M_{left} - S + 2t > S$$

$$2M_{left} - S + 2t > S$$

$$2M_{left} > 2S - 2t$$

Set $M_{left} := S - t + 1$

Subset of $\langle a_1, \dots, a_n \rangle$ adding to t $\Longleftrightarrow$ even partition of $\langle a_1, \dots, a_n, M_{left}, M_{right} \rangle$

//

Space-bounded computation

If TM M halts on input w,

Then it uses space = index of leftmost cell that is never scanned.

PSPACE $:= \{ L(M) :$ M is a decider that uses space Poly(n) for inputs of length n $\}$

Easy: P $\subseteq$ PSPACE

"Easy": NP $\subseteq$ PSPACE