WHAT HAPPENS IN CASES WHERE WE ARE UNABLE TO FIND EFFICIENT SOLUTIONS TO ALGORITHMIC PROBLEMS?

SOME OF THE PROBLEMS WE STUDIED

① AN ALGORITHM FOR FINDING A SHORTEST PATH IN GRAPHS WITH POSITIVELY-WEIGHTED EDGES. $(O((n+m)\cdot \lg n))$

② AN ALGORITHM FOR INTERVAL SCH. $(O(n \lg n))$

...

ALSO, WE STUDIED SOME OF THEIR "GENERALIZATIONS":

③ AN ALGORITHM FOR FINDING SHORTEST PATH IN ARBITRARY GRAPHS: $(O(n \cdot m))$

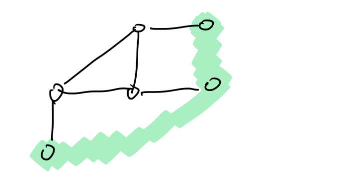
④ AN ALGORITHM FOR WEIGHTED INTERVAL SCHEDULING.

...

OTHER GENERALIZATIONS:

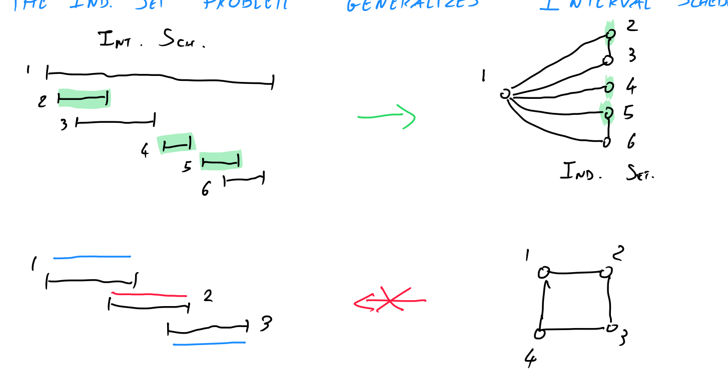
⑤ INDEPENDENT SET

GIVEN AN UNDIRECTED GRAPH $G(V,E)$, WE SAY THAT $S \subseteq V$ IS AN INDEPENDENT SET OF $G(V,E)$ IF THERE ARE NO EDGES OF $G(V,E)$ AMONG THE NODES OF S.



THE IS PROBLEM ASKS FOR A LARGEST IS IN THE GRAPH.

THE IND. SET PROBLEM GENERALIZES INTERVAL SCHEDULING



IF WE COULD FIND AN ALG. FOR IND. SET WE WOULD HAVE A NEW SOLUTION FOR INT. SCH.

REDUCTIONS

REDUCTIONS ARE, MAYBE, THE MOST IMPORTANT TOOL IN CS. THEY ARE BASED ON THE FOLLOWING IDEA:

- SUPPOSE THAT THERE EXISTS AN ORACLE (A BLACK BOX, OR A FUNCTION) THAT IN POLYTIME, SOLVES INSTANCES OF PROBLEM X.
- SUPPOSE, ALSO THAT YOU DEVISE A REDUCTION (AN ALGORITHM) THAT, WITH POLYNOMIALLY MANY CALLS TO THE ORACLE FOR X, PLUS SOME EXTRA POLYTIME COMPUTATION, SOLVES INSTANCES OF PROBLEM Y.

THEN, YOU CAN SOLVE INSTANCES OF Y IN POLYNOMIAL TIME.

DEF: FOR A PAIR OF PROBLEMS X AND Y, WE WRITE $Y \leq_p X$ IF THE ABOVE REDUCTION EXISTS.

INTERVAL SCHEDULING $\leq_p$ SORTING
INTERVAL SCHEDULING $\leq_p$ WEIGHTED INT. SCHEDULING

IF WE HAVE "$A \leq_p B$" THEN:

- IF B CAN BE SOLVED EFFICIENTLY, THEN A CAN ALSO BE " " .
- IF A CANNOT BE SOLVED EFFICIENTLY, THEN B ALSO " " .

L1: IF $Y \leq_p X$, AND X CAN BE SOLVED IN POLYTIME, THEN Y CAN ALSO BE SOLVED IN " ".

P: EACH CALL TO THE X-ORACLE TAKES POLY-TIME. THAT IS $\exists c \geq 0$ S.T. FOR EACH INSTANCE I OF PROBLEM X THE CALL ORACLE(I) TAKES TIME AT MOST $O(|I|^c)$.

THE REDUCTION (GIVEN BY $Y \leq_p X$) MAKES AT MOST $n^{c'}$ CALLS TO THE ORACLE, EACH FEEDING AN INPUT TO THE ORACLE OF SIZE AT MOST $O(n^{c''})$ (FOR SOME $c', c'' > 0$).

THE REDUCTION MIGHT ALSO MAKE SOME EXTRA POLY TIME COMPUTATION — SAY $O(n^{c'''})$ FOR $c''' > 0$.

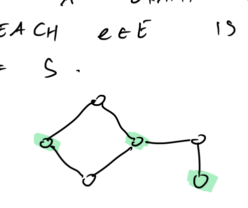
THE TOTAL RUNTIME IS THEN:

$$O(n^{c'''}) + O(n^{c'}(n^{c''})^c) = O(n^{\max(c''', c' + c \cdot c'')}). \square$$

L2: IF $Y \leq_p X$, AND Y CANNOT BE SOLVED IN POLYTIME, THEN X " " " " ".

VERTEX COVER (AKA "GUARDS IN A MUSEUM")

A VC OF A GRAPH $G(V,E)$ IS A SET $S \subseteq V$ SUCH THAT EACH $e \in E$ IS INCIDENT ON AT LEAST ONE NODE OF S.



THE VC PROBLEM ASKS FOR A <u>SMALLEST</u> VERTEX COVER OF $G(V,E)$.

L: INDEPENDENT SET $\leq_p$ VERTEX COVER
L: VERTEX COVER $\leq_p$ INDEPENDENT SET

T: LET $G(V,E)$ BE AN UNDIRECTED GRAPH.
THEN, $S \subseteq V$ IS AN IND. SET OF $G(V,E)$ IFF $V-S$ IS A VERTEX COVER OF $G(V,E)$.

P: FIRST, ASSUME THAT S IS AN IND. SET.
$\forall \{u,v\} \in E$ EITHER $u \notin S$, OR $v \notin S$, OR $u, v \notin S$.
BUT, THEN, $V-S$ MUST BE A VERTEX COVER (INDEED, $\forall \{u,v\} \in E$ EITHER $u \in V-S$, OR $v \in V-S$, OR $u, v \in V-S$).

SECOND, SUPPOSE THAT $V-S$ IS A VC. THEN IF $u, v \notin V-S$, $u \neq v$, THEN $\{u,v\} \notin E$.

NOW, $u, v \notin V-S$ IS EQUIVALENT TO $u, v \in S$.
THEREFORE, S IS AN IND. SET. $\square$

THEN:

C: $\min VC \leq k \iff \max \text{Ind Set} \geq n - k$

THAT IS, VC AND IND SET ARE EQUIVALENT W.R.T. POLY TIME REDUCTIONS.

TRANSITIVITY OF REDUCTIONS

L: IF $Z \leq_p Y$ AND $Y \leq_p X$ THEN $Z \leq_p X$.

P: WE AIM TO SOLVE INSTANCES OF Z USING AN ORACLE FOR X.

FIRST, WE RUN THE REDUCTION FOR SOLVING INST. OF Z BY SOLVING POLINOMIALLY MANY INST. OF Y.

SECOND, WE RUN THE REDUCTION FOR SOLVING INST. OF Y BY SOLVING INSTANCES OF X.

THIS GIVES US A POLYNOMIAL REDUCTION FOR $Z \leq_p X$. $\square$

INTERVAL SCHEDULING $\leq_p$ IND SET AND
IND SET $\leq_p$ VC ENTAIL
INT. SCH. $\leq_p$ VC.

WHAT IS P?

DEF: P IS THE SET OF DECISION PROBLEMS X SUCH THAT $\exists \gamma \geq 0$ AND AN ALG. A S.T. FOR EACH INPUT I OF X, WITH $|I| = n$, A CORRECTLY DETERMINES IN TIME $O(n^\gamma)$ (IN POLYTIME) IF I IS A "YES"-INSTANCE OR A "NO"-INSTANCE OF X.

X = "DOES THIS INST. OF INT. SCH. ADMIT A SOLUTION WITH AT LEAST $\frac{n}{2}$ INTERVALS?"

THEN, $X \in P$.

"CERTIFICATES"

SOMETIMES, IT IS HARD SOLVE A PROBLEM, BUT IT IS EASY TO CHECK WHETHER A SOLUTION IS VALID.

- IN VERTEX COVER, IF YOU GIVE ME $S \subseteq V$, I CAN EASILY CHECK WHETHER S IS A VC.

  THE QUESTION "DOES THERE EXIST A VC OF SIZE AT MOST k?" ADMITS A SIMPLE CERTIFICATE (YES-CERTIFICATE): A SET OF NODES.

LET US GIVE THE DEFINITION OF A CERTIFIER.

DEF: LET $\gamma \geq 0$ BE A CONSTANT.
AN ALGORITHM B (TAKING AS INPUT TWO STRINGS: AN INSTANCE I AND A CERTIFICATE C OF, RESPECTIVELY, $|I| = n$ BITS AND $|C| \leq n^\gamma$ BITS) IS AN EFFICIENT CERTIFIER FOR X IF:

(i) THE RUNTIME OF B IS AT MOST $O(n^\gamma)$;
(ii) I IS A "YES"-INSTANCE OF X IFF $\exists$ A CERTIFICATE C S.T. $B(I,C) = $ TRUE;
(iii) I IS A "NO"-INSTANCE OF X IFF $\forall$ CERTIFICATE C: $B(I,C) = $ FALSE.

DEF: NP IS THE CLASS OF PROBLEMS THAT ADMIT AN EFFICIENT CERTIFIER.

NP STANDS FOR "NON-DETERMINISTIC POLYTIME"
(NP DOES NOT STAND FOR "NON-POLYTIME")

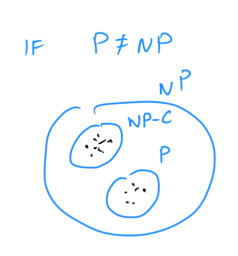
$P \stackrel{?}{=} NP$ $\quad 10^6$ \$

L: $P \subseteq NP$

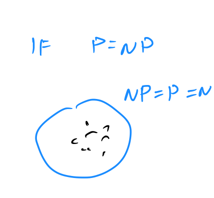
P (SKETCH): $\forall X \in P$, ONE COULD CREATE A "CERTIFIER" THAT THROWS AWAY THE CERTIFICATE AND SOLVES THE INSTANCE IN POLYTIME.

Q: IS $P = NP$?

IF $P \neq NP$



IF $P = NP$



D: IF $X \in NP$, AND $\forall Y \in NP$ IT HOLDS $Y \leq_p X$, THEN X IS NP-COMPLETE (NPC).

COOK-LEVIN THEOREM

INDEPENDENT SET <u>IS</u> NP-COMPLETE.
$(\forall Y \in NP: Y \leq_p \text{Ind Set})$

$\Downarrow$

VERTEX COVER IS NP-COMPLETE
$(\forall Y \in NP: Y \leq_p \text{Ind Set})$ AND $(\text{Ind Set} \leq_p VC)$

$\Downarrow$ TRANS. OF REDUCTIONS

$\forall Y \in NP: Y \leq_p VC$ (VC IS NP-COMPLETE)

L: $VC \leq_p$ KNAPSACK $\implies$ KNAPSACK IS NP-COMPLETE

L: IND. SET $\leq_p$ "LONGEST PATH IN A GRAPH" ($\in$ NP-COMPLETE)