Section 10: P/NP

$A \in NP\text{-hard}$

$B \in NP$

$A \le_p B$

0. Definitions

$\forall C \in NP \quad C \le_p A \Rightarrow C \le_p B$

a) What does P stand for?

b) What is NP stand for?

c) What is the definition of P?

$f \in P$ iff f has poly time algo.

d) What is the definition of NP?

$f \in NP$ iff $\forall x$ s.t. $f(x) = 1$
$\exists$ "proof" y and fast verification given y.

1. P & NP Membership

a) How would we show that a given problem belongs to the class P?

give fast algo.

b) How would we show that a given problem belongs to the class NP?

NP-complete

everything in NP reduces to A,

and A is in NP

NP-hard

everything in NP reduces to A.

$f \in P$. given $x \in f^{-1}(1)$ specify proof y, and fast verification algo

2. P & NP Membership

For problems A and B below, show that they belong to both P and NP. Show that problem C belongs to NP.

a) Problem A: Given a list of 2-dimensional points, return true or false to indicate whether some pair of points have a distance of less than 5.

Belongs to P

```
for x in L:
    for y ≠ x in L:
        if dist(x,y) ≤ 5:
            return true.
return false.
```

Belongs to NP

1) $f \in P \subseteq NP$.

2) L is list, $dist(L_i, L_j) \leq S$.

proof y: index $i \in [n]$, index $j \in [n]$.

b) Problem B: Given a list of integers, return true or false to indicate whether any items are duplicated (so return true if some value appears at least twice, false if all are distinct).

Belongs to P

Belongs to NP

c) Problem C: Given a weighted graph, a pair of nodes X and Y, and a number k, return true or false to indicate whether there is a path from X to Y with a cost of at least k

Belongs to NP

"proof" : a path

verification:

① check every edge exists on this path

② path starts at X, ends at Y

③ $w(\text{path}) \geq k$.

$$\Phi = (x_1 \vee x_3 \vee \bar{x}_5) \wedge (x_7 \vee \bar{x}_{12} \vee x_8)$$

...

"3CNF"

$A = \{\Phi : \Phi$ does NOT have a satisfying assignment$\}$

$$\forall x_1 \forall x_2 \forall x_3 \forall x_4 \ldots \forall x_n \cancel{\neg} \Phi(x_1, \ldots, x_n).$$

$\exists \quad \exists \quad \exists \quad \exists \quad \exists.$

3. NP-Hard and NP-Complete Definitions

a) What is the definition of NP-Hard?

b) What is the definition of NP-Complete?

4. NP-Hard and NP-Complete Membership

a) How do you show that a problem C belongs to NP-Hard

some NP-complete problem B reduces to it

or hard

$\forall A \in NP,\ A \leq_p B \leq_p C$

b) How do you show that a problem belongs to NP-Complete

some NP-complete problem B reduces to it

or hard

$\forall A \in NP,\ A \leq_p B \leq_p C$

+ C must be in NP itself

$\forall B \in NP \quad B \leq_p C \leq_p A \implies B \leq_p A$

$f \in NP\text{-hard}$ $\quad f \leq_p A$

5. Practice

$A \leq_p B$

If A polynomial-time reduces to B and B is NP-Hard then A is NP-Hard.

True False

If B is NP-Hard and there exists a polynomial time algorithm for B, then P=NP.

True False

If B is NP-Hard and there does not exist a polynomial time algorithm for B, then P does not equal NP.

True False

If A reduces to B in polynomial time, and B reduces to C in polynomial time, and A is NP-Hard, then C is NP-Hard.

$\forall f \in NP,\ f \leq_p A \leq_p B \leq_p C$

True False

If A reduces to B in polynomial time, and B reduces to C in polynomial time, and A is NP-Complete, then C is NP-Complete.

True False

If A reduces to B in polynomial time, and B reduces to C in polynomial time, and A is in EXP, then C is EXP.

True False

If A reduces to B in polynomial time, and B reduces to C in polynomial time, and A is in P, then C is P.

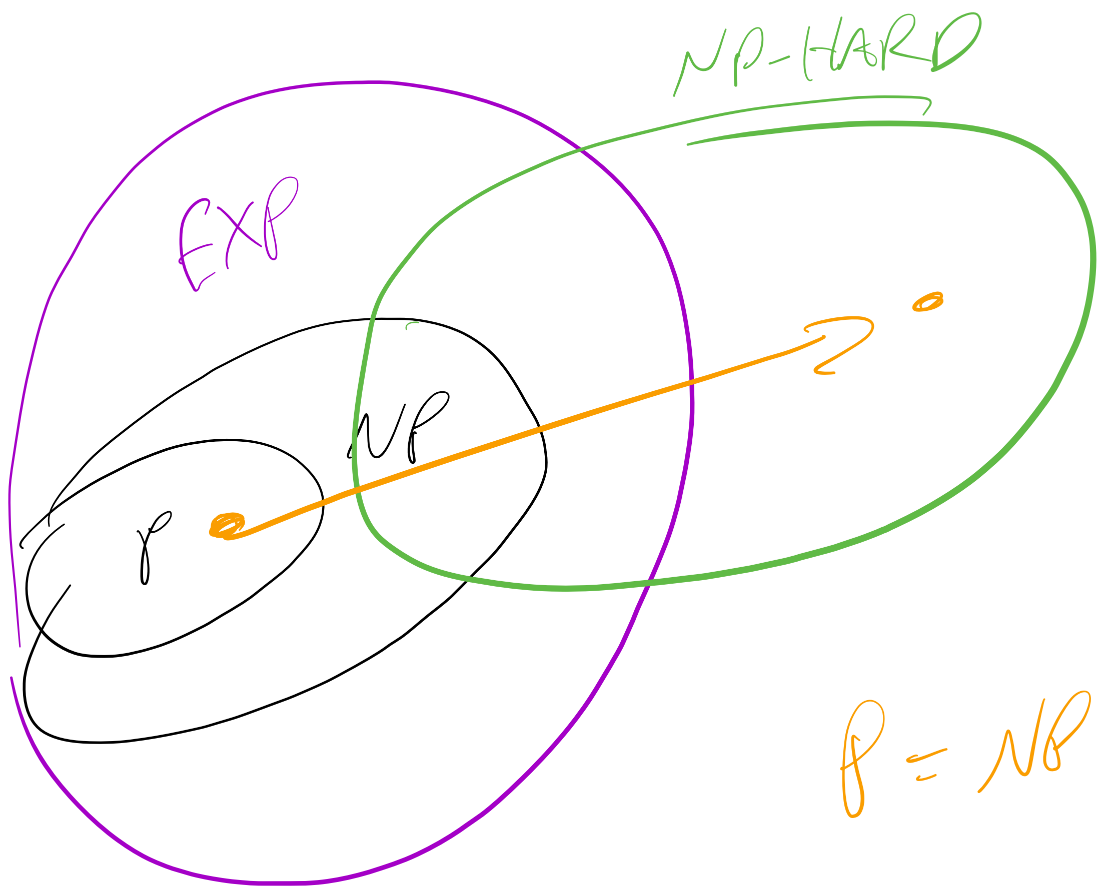
True False

If A reduces to B in polynomial time, and B reduces to C in polynomial time, and A is in NP, and C is in P, then P=NP.

True False

If A reduces to B in polynomial time, and B reduces to C in polynomial time, and A is in NP-Hard, and C is in P, then P=NP.

True False


NP-HARD
EXP
NP
P
?
P = NP