[redacted]

1) P: the set of decision problems that can be solved by a Turing machine in Polynomial time w.r.t. input

NP: The set of decision problems whose candidate answers can be verified in Polynomial time.

NP Hard: The set of decision problems that are as hard as any problem in NP (ie. any problem in NP can be reduced to an instance of an NP Hard Problem).

NP complete: The set of decision problems which are in NP and are NP-Hard.

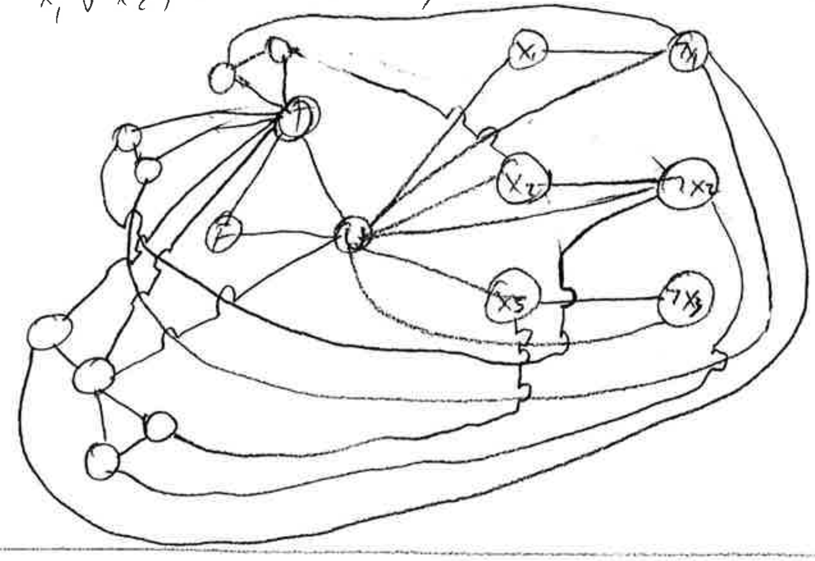
NP-complete if P = NP

- Dot Product is in P and NP, as well as NC
- Interval computations is NP-Hard | P is P = NP

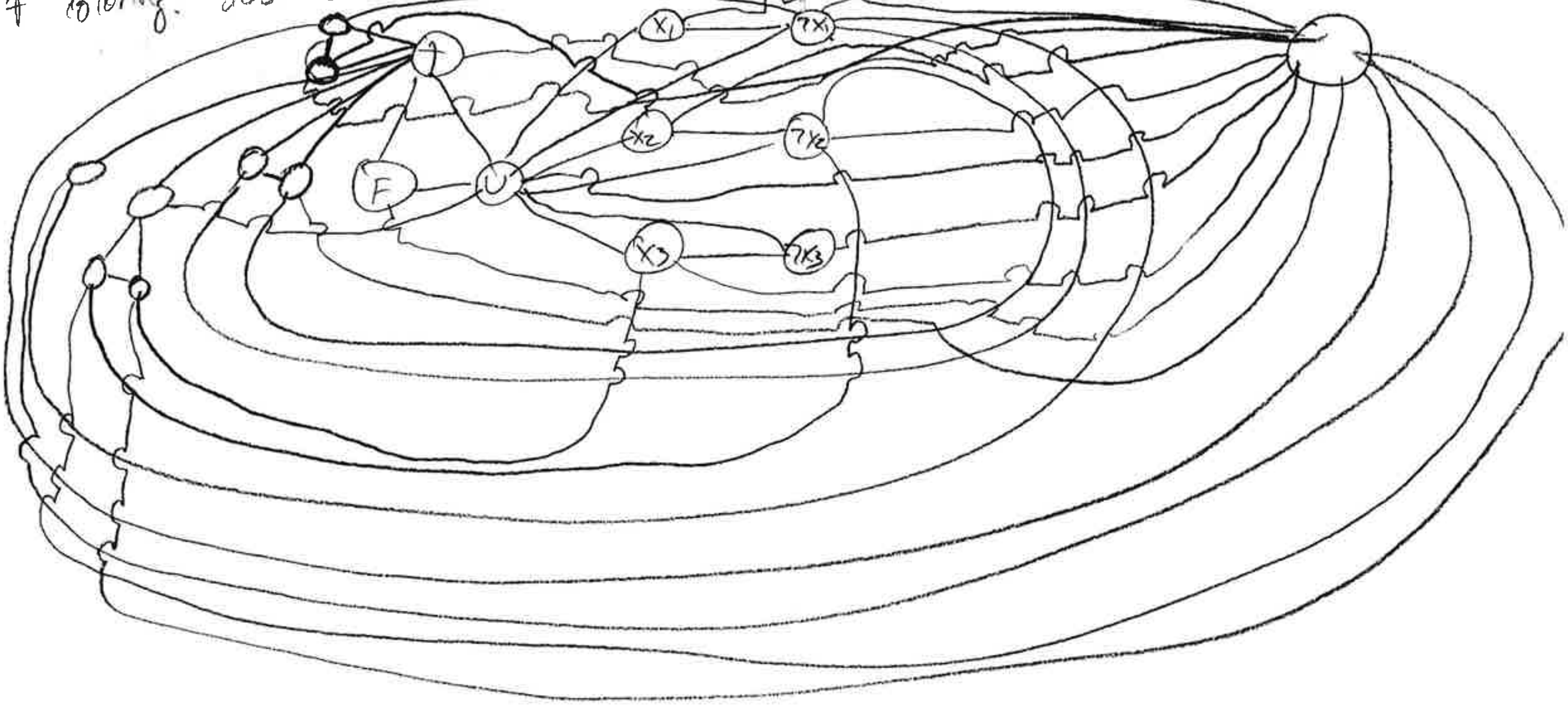
[redacted]

2) $F = (\neg x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$

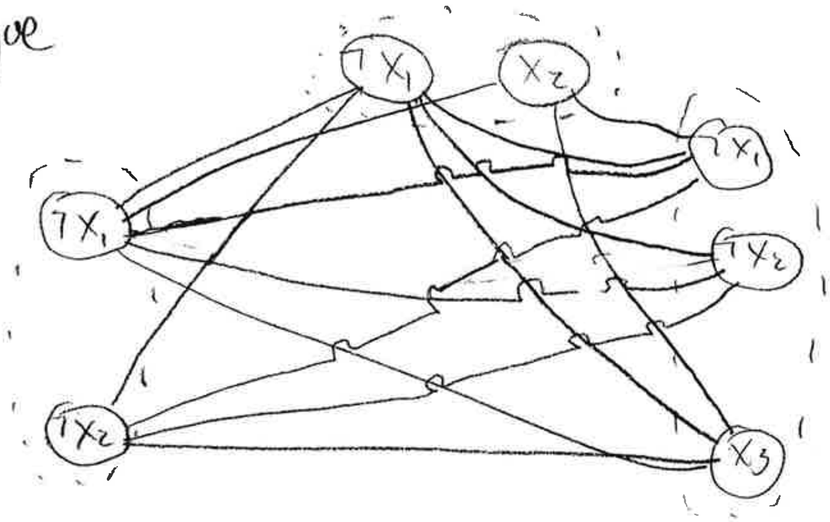
3 coloring



4 coloring: Just add one more node (color) and connect all previous nodes to it.



Clique



Subset Sum Problem

	1	2	3	C1	C2	C3
x_1	1	0	0	0	0	0
$\neg x_1$	1	0	0	1	1	1
x_2	0	1	0	1	0	0
$\neg x_2$	0	1	0	0	1	1
x_3	0	0	1	0	0	1
$\neg x_3$	0	0	1	0	0	0
g_1	0	0	0	1	0	0
h_1	0	0	0	1	0	0
g_2	0	0	0	0	1	0
h_2	0	0	0	0	1	0
g_3	0	0	0	0	0	1
h_3	0	0	0	0	0	1
	1	1	1	3	3	3

Interval Computation

$$F(x_1, x_2, x_3) = \left(1-(x_1)(1-x_2)\right)\left(1-(x_1)(x_2)\right)\left(1-(x_1)(x_2)(1-x_3)\right)$$

[redacted]

7-9

- Yes, it belongs to NC (Nick's class) because we can solve the problem in polylogarithmic time on a parallel computer with a Polynomial number of processors. this is a definition not an explanation

- Is $NC \subseteq P$? Yes. We just simulate all polylogarithmic parallel computations sequentially, and this is still done in Polynomial time.

- Is $NC = P$? We don't know, but if we can effectively parallelize a P-hard problem such as linear programming and solve it in polylogarithmic time using a Polynomial number of processors, then $NC = P$

- If we parallelize & take into account communication time, what's the fastest we get?

Taking into account communication time:

$R = c \cdot T_{par}(n)$
↑
speed of light

$$V = \frac{4}{3}\pi R^3 = \frac{4}{3}\pi c^3 T^3_{par}(n)$$

$$N_{proc} \le \frac{V}{\Delta V} = \left(\frac{4}{3}\cdot\frac{\pi c^3}{\Delta V}\right)\cdot\left(T^3_{par}(n)\right)$$

$$T_{seq}(n) \le \text{Total time} = T_{par}(n) \cdot N_{proc} \le \text{const} \cdot T_{par}(n) \cdot T^3_{par}(n)$$

$$T_{seq}(n) \le c \cdot T^4_{par}(n)$$

$$c \cdot T_{seq}^{1/4} \le T_{par}(n)$$

what?

- Similar arguments are used in the proof that SAT is NP-Hard, the speed of light as the fastest we can go

① $V \le C$, we are bounded by the speed of light as the fastest we can go

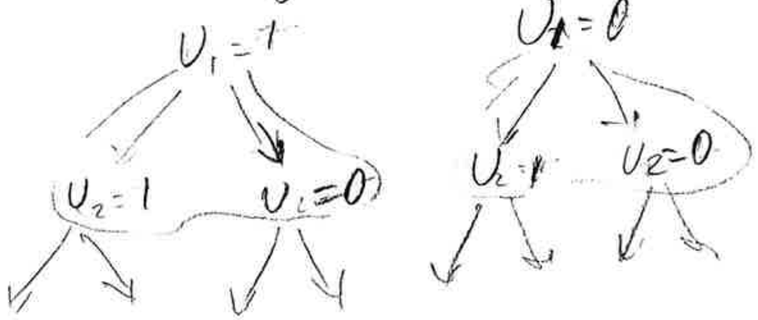
② The sphere's max Volume is bounded by Euclidean space.

[redacted]

• Explain the physics...

The maximum speed we can travel by is the constant c, the speed of light. If we were to travel with no bound on speed, we could just go as fast as we want. Another scenario would be to effectively travel in time as in a time machine, and send signals to our time when needed.

The Volume, and the amount of processors we can fit inside the sphere, is bounded by Euclidean physics. If we were to violate this, and use Lobachevskys space, we could fit an exponential number of processors in the sphere, and just run many more instances in parallel:



Almost Black hole: Consider sat. The answer is sent back in linear time! The depth of the tree is proportional to n (problem size)

$\frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4}$

10) $P_{error} = \frac{1}{4}$

$\varepsilon = 0.1\% = 0.001 = 10^{-3}$

$\frac{1}{4}^k \leq 10^{-3}$

$4^k \geq 10^3$

$4^k \geq 1000$

$(2^2)^k \geq 1000$

$2^{2k} \geq 2^{10}$

$2k \geq 10$

$k \geq 5$

or $k = \frac{\ln P_{err}}{\ln P}$

11) $\Pi_2 P^{\Sigma_4 P} = \forall x \underbrace{\exists y \exists a}_{\text{combine}} \forall b \exists c \forall d \; P$

$\Pi_5 P$

12) $\exists x \forall y \exists z \; win(x, y, z, t) \equiv \boxed{\Sigma_3 P}$

13)

```
for(int i=0; i< 2014; i++)
    System.out.print("100");
```

46 characters counted

$K(x) \leq 46$