CSE525 Lec17
Reduction

•••

Debajyoti Bera (M21)

Mod3Path

def MOD3PATH(G, s, t):
(H, s', t') ← Reduce(G, s, t)
Run BFS(H, s', t')
If s' $\overset{H}{\rightsquigarrow}$ t', return yes
else return no

→ Input is a Yes instance → answer is yes/True
Decision problem: " No " → answer is No/False

Given a graph G and two vertices s and t, is there a path from s ⇝ t of length divisible by 3 ?

def Reduce(G,s,t):

H: for every v ∈ G, add (v,0), (v,1), (v,2) to H
add edges (see tutorial notes)

// construct and return (H,s',t') such that ...

Reduction Lemma

// (G,s,t) is a **YES** instance of Mod3Path iff (H,s',t') is a **YES** instance of REACHability

REACH (H, s', t'): Yes if s' ⇝ t' in H
No if no such path

Graph, two vertices → Yes/No

- Input and output of Mod3Path ?
- Input and output of Reduce ? input: instance of MOD3PATH (Graph + two vertices)
  output: " " REACH (Graph + " ")
- Time complexity of "reduction" ?
  ↳ in terms of input to reduce

Reduction (for decision problems)

decision decision

$P \leq_p Q$: (many-one polynomial-time) reduction of problem P to problem Q

converting P's instances to Q's instances

- An algorithm to convert any instance/input X of P to an instance/input Y of Q
- Running time of reduction algorithm : poly(size of X)
- P(X) returns True iff Q(Y) returns True Reduction lemma

X is a Yes instance of P iff

Y=Reduce(X) is a Yes instance of Q.

Not about algorithm for solving P(X)

def Reduce(A[1⋯n]):
return {(1,0), (2,0), (3,0) }

3SUM reduces to Collinearity

3SUM: Given array A of integers, does it contain a,b,c whose sum is zero?

COLLINEARITY: Given a set of points in 2D, does it contain 3 points which lie on same line?

$3SUM \leq_p COLL$

Instance of 3SUM: Array A of integers

Instance of COLL: set of 2D points

def 3SUMtoCOLL-Try1(A):

ⓐ holds Return S = {(x,1) : x in A}

ⓑ Take A=[1,2,3] S={(1,1),(2,1),(3,1)} ✗

ⓐ If A has a 3SUM solution then S has 3 coll. points.

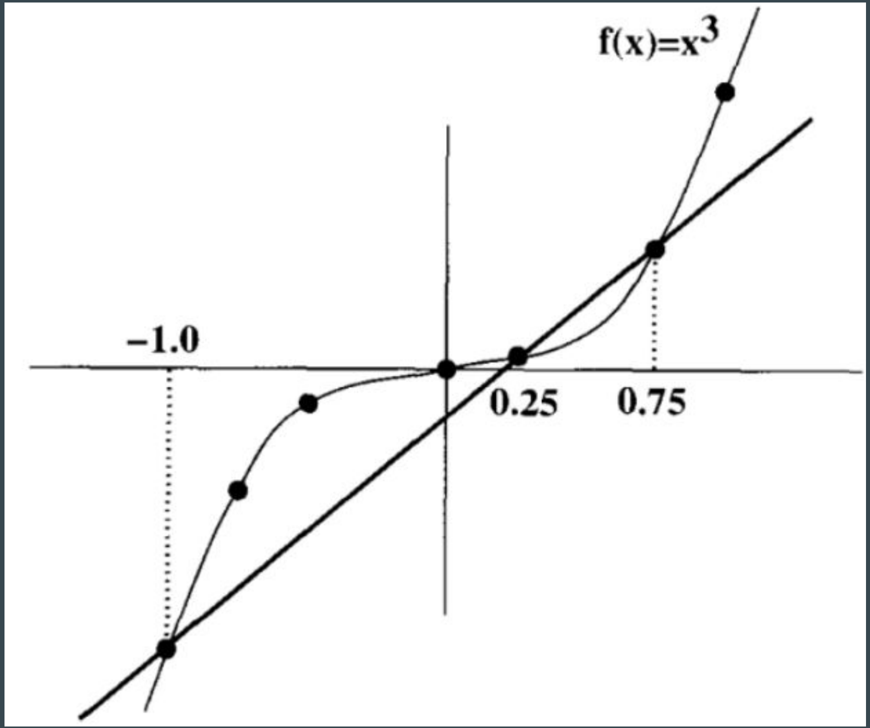
+ ⓑ If S has 3 coll points then A has a 3SUM solution.

$\boxed{\text{A has a 3SUM solution iff S=Reduce(A) has 3 collinear points.}}$ Is this true?

Reduce([1,2,3]) → {(1,0), (2,0), (3,0)}

ⓐ is satisfied (False ⇒ True)

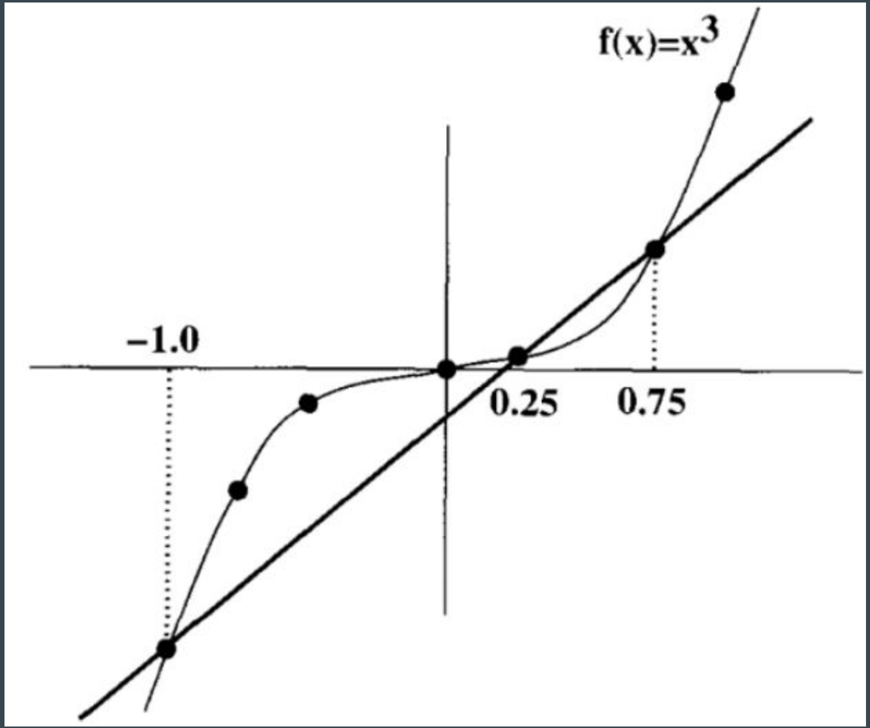
ⓑ (True ⇒ False) is not satisfied



3SUM reduces to Collinearity

3SUM: Given array A of integers, does it contain a,b,c whose sum is zero?

COLLINEARITY: Given a set of points in 2D, does it contain 3 points which lie on same line?



Instance of 3SUM: Array A of integers

def 3SUMtoCOLL(A):

Return S = {(x,x) : x in A}

A has a 3SUM solution iff S=Reduce(A) has 3 collinear points. Is this true?

ⓐ S is always collinear. ∴ It will be coll. even when A has a 3SUM solution.

ⓑ Doesnot hold. A = [1, 2, 3] S = Reduce(A) = {(1,1), (2,2), (3,3)}

S has 3 coll. points but A has no 3SUM soln.

```
def 3SUMtoCOLL(A):
    Return S = {(x,x^3) : x in A}
```

A has a 3SUM solution iff S=Reduce(A) has 3 collinear points. Is this true?

Complexity : $O(|A|)$
of 3SUMtoCOLL

(a) Take any A with a valid 3SUM solution.
$\exists\, a,b,c \in A$ s.t. $a+b+c=0$
S has the points $\underset{p_1}{(a,a^3)}, \underset{p_2}{(b,b^3)}, \underset{p_3}{(c,c^3)}$
Claim: These points are collinear.

Ex. Prove that $\dfrac{b^3-a^3}{b-a} = \dfrac{b^3-c^3}{b-c}$ when $a+b+c=0$

(b) Take any A s.t. S=Reduce(A) has 3 collinear points. Let those points be $(x,x^3), (y,y^3), (z,z^3)$

Since they are coll.; $\dfrac{y^3-x^3}{y-x} = \dfrac{z^3-x^3}{z-x}$

Ex. Prove that $x+y+z=0$

LIS $\leqslant$ LCS

Exercise

LIS(A,k): length of the longest increasing subsequence in A has length k

LCS(B,C,m) : length of the longest common subsequence in B and C has length m

```
def LIStoLCS(A,k):
    ???
    return (B,C,m)
```

Lemma: LIS(A,k) is true <u>if and only if</u> LCS(B,C,m) is true.

(a) If G can be coloured using 3 colours then Reduce(G) can be coloured using 4 colours.
(b) If Reduce(G) " " " 4 " " G " " " " 3 colours.

3COLOR $\leqslant_p$ 4COLOR

$\Updownarrow$

(b) If G cannot be coloured using 3 colours, then Reduce(G) can't be coloured using 4 colours.

3COLOR(G): Can G be coloured using at most 3 colours?

4COLOR(G): Can G be coloured using at most 4 colours?

Q: Reduce 3COLOR to 4COLOR.

input to reduction: graph
output of reduction: graph

def 3COLto4COL(G):

// Try 1: return G (a) holds, (b) does not hold

// Try 2: colour G using 3 colours. ~~Then what?~~

If we find a colouring output , if not

// Try 3: ???

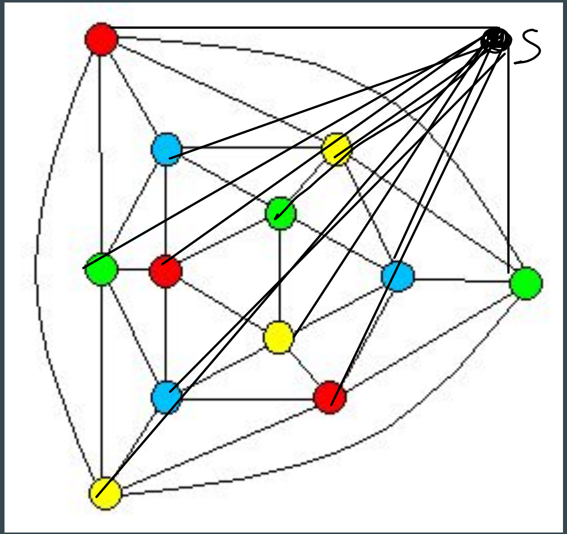


Image: mathpages.com

Lemma: G can be coloured using 3 or less colours iff H=3COLto4COL(G) can be coloured using 4 or less colours.

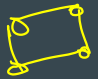
Try 2 Reduce() = 

Try 2

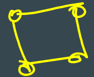
def Reduce(G): → n vertices, m edges

→ for all possible assignment of 3 colours to vertices of G:

→ if the colouring is valid:

output 

output 

(a) holds [If G has a valid 3colouring Reduce(G) =  can be coloured using 4 colours]

(b) If G cannot be coloured using 3 colours, Reduce(G) =  can't be coloured using 4 colours.

Complexity: $O(3^n \times m)$ Not poly($|G|$)

Try 3: Construct H which is a copy of G. Add a new vertex s to H & add edges from s to all other vertices. return H. (a) holds (b) Think?