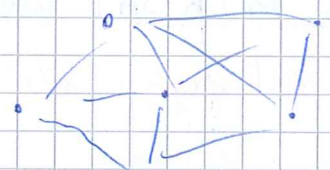
Complexity of Polynomial-Time Problems

Motivation

Idea: Study ~~the~~ Structure and Hardness ~~in~~ within the complexity class P (polynomial time)

Informally: Answer the question why some algorithms are essentially as fast as possible

Example: All-Pairs Shortest Path in a weighted graph



("pairwise distances")

n nodes, m edges

Textbook solution: $O(n^3)$ time, Floyd Warshall

Sparse graphs: $O(mn + n^2 \log n)$, Dijkstra

Question: How fast can we solve ~~most~~ problem when $m \approx n^2$ (dense graphs)

$O(n^3)$ 1962

$O(n^3/\log^{1/3} n)$ 1975

$O(n^3/\log^{1/2} n)$ 1990

$\vdots$

$O\left(n^3 / 2^{\Omega(\sqrt{\log n})}\right)$ 2014

But: no $O(n^{3-\delta})$-time algorithm (for constant $\delta > 0$) known, not even $n^{2.99}$

Deeper ~~R~~ reason? ~~Yes~~ Yes!

$n^{3-\delta}$-alg for APSP $\Leftrightarrow$ $n^{3-\delta}$ alg for ~~betweenness centrality~~ radius of graph

$\Leftrightarrow$ $n^{3-\delta}$ alg for finding neg. triangle

$\Leftrightarrow$ ... (10-15 more)

Example: Longest Common Subsequence of two strings

A G C A T   length $\leq n$

G A C

$O(n^2)$ [Aho et al '76]

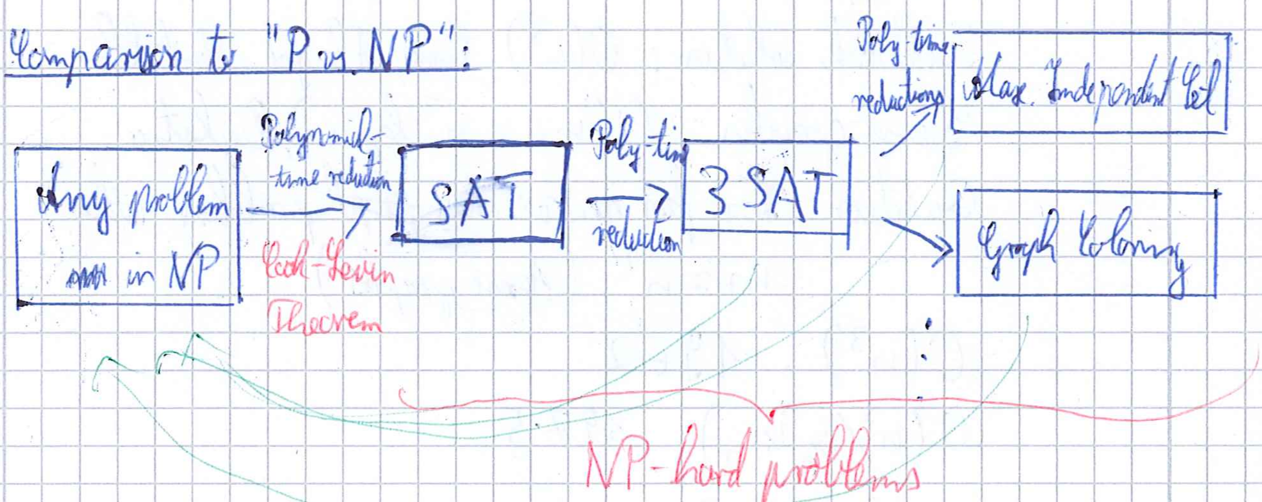
$O(n^2/\log n)$ [Masek/Paterson '80]

Deeper reason?

If there is an $n^{2-\delta}$-time alg. for LCS, then there is a $2^{(1-\delta)N}$ algorithm for k-SAT with N variables

Would be a major breakthrough in SAT solving

Comparison to "P vs NP":



NP-complete: in NP + NP-hard

The reductions show: If one NP-complete problem has a polynomial-time algorithm, then all of them have

Our Goal (mainly): Prove conditional lower bounds for problems in P based on conjectured hardness of a few, central problems

In P: time complexity of reductions matters, polynomial time is not enough!

~~Relating~~ From Orthogonal Vectors ~~and~~ to NFA Acceptance

~~Input~~ Definition: Orthogonal Vectors Problem:

Input: Sets $A, B \subseteq \{0,1\}^d$ of size n

Task: Decide if there are ~~a, b~~ $a \in A$, $b \in B$ s.t. $a \perp b$

$$\langle a, b \rangle = 0 \iff \sum_{i=1}^{d} a[i]b[i] = 0$$

$$\iff \forall 1 \le i \le d : a[i] = 0 \text{ or } b[i] = 0$$

Example: $A: (1,1,1)\ (1,1,0)\ (1,0,1)\ (0,0,1)$

$B: (0,1,0)\ (0,1,1)\ (1,0,1)\ (1,1,1)$

Algorithm: Time $O(n^2 d)$ – trivial

Fastest ~~#~~ known algorithm: Time $O(n^{2-1/O(\log c)})$ if $d = c \log n$

Most interesting case: $d = \Theta(\log n)$

Open Question:

Can we do substantially better?
(e.g. $O(n^{1.99})$)

From Orthogonal Vectors to Diameter

Def.: Diameter Problem

Input: (Unweighted) graph $G=(V,E)$, $n=|V|$, $m=|E|$

Task: Compute longest distance between any pair of vertices

$$= \max_{u,v\in V} \text{dist}_G(u,v)$$

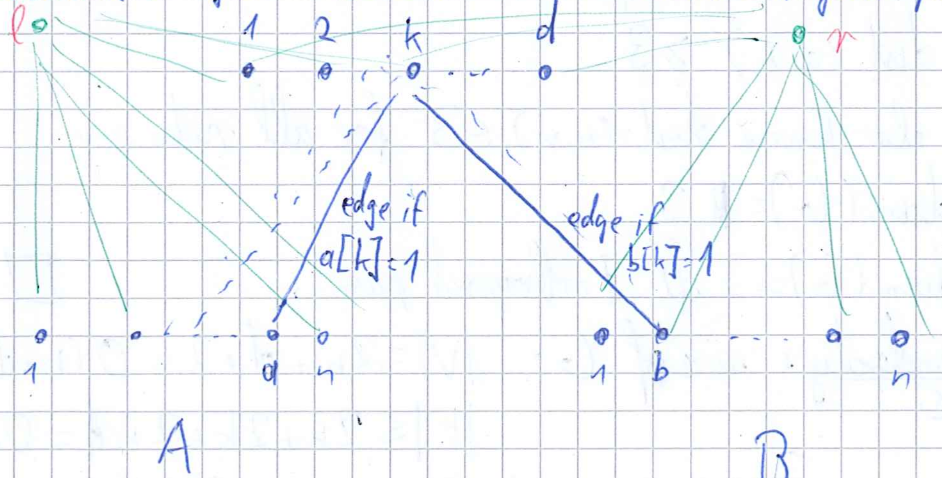
(distance = length of the shortest path)

Simple algorithm:
- single source shortest paths: $O(m)$ (unweighted) BFS, $O(m+n\log n)$ weighted Dijkstra ~~Dijkstra BFS~~
- all-pairs shortest paths: $O(mn)$, $O(mn+n^2\log n)$
  SSSP from every node
- suffices to compute diameter in additional $O(n^2)$ time

Theorem: Assuming OVH, there is no $O((mn)^{1-\delta})$-time algorithm (with constant $\delta>0$) ~~unless~~ for computing the diameter of a graph.

Proof: Preprocessing Step: make sure, every vector has at least one '1'-entry
Otherwise: output that there is orthogonal pair
Time $O(nd)$

Reduction: Goal: diameter $= 2 \iff$ ~~xxx~~ no orthogonal pair



every $a\in A$ has some k s.t. $a[k]=1$
thus $\exists$ path a-k-r-b

$\text{dist}(u,v) \le \cancel{4}3$ for all pairs u,v also $\text{dist}(a,b)\ge 2$ because no direct edge over elements

More precisely: $\text{dist}(u,v)\le \cancel{4}3$ if $u\in A$ and $v\in B$

$\text{dist}(u,v)\le 2$ otherwise

$\Rightarrow$ only have to further analyze $\text{dist}_G(a,b)$ $a\in A$, $b\in B$

- If there is no orthogonal pair, then for every $a \in A$, $b \in B$ there is a k s.t. $a[k] = 1$ and $b[k] = 1$

$$\Rightarrow \exists \text{ path } a - k - b$$
$$\Rightarrow dist(a,b) \leq 2$$
$$\xRightarrow{dist(a,b) \geq 2} dist(a,b) = 2$$

As for every other pair of nodes, $dist(u,v) \leq 2$, we get $diam(G) = 2$

- If there is an orthogonal pair, then there are $a \in A$, $b \in B$ s.t. $a \perp b$, i.e. whenever $a[k] = 1$, we have $b[k] = 0$

No path from a to b has length < 3

Enumerate all possibilities (simple paths):
  - $a - l - a' - k$ for some k
  - $a - l - k - r$ for some k
  - $a - l - k - b'$ ~~(if it exists)~~ (for some k, b')
  - $a - k - a' - l$ (for some k, a')
  - $a - k - a' - k'$ (for some k, k')
  - $a - k - r - k'$ (for some k, k')
  - $a - k - r - b'$ (for some k, b')

$\Rightarrow dist(a,b) \geq 3$

We also know $dist(u,v) \leq 3$ for all nodes u, v

$\Rightarrow diam(G) = 3$

Thus: $diam(G) = 3$ iff $\exists$ orthogonal pair $\square$

<u>Time Complexity</u>: ~~Observation:~~ size of G: $|V| = 2n + d + 2 = O(n+d)$

$|E| \leq 2n + 2k + 2nd = O(nd)$

Time for constructing $G = O(nd)$

Suppose $O((|E||V|)^{1-\delta})$-time alg for diameter

$\Rightarrow$ Have alg. for OV with running time $O((n \cdot d \cdot (n+d))^{1-\delta}) = O(n^{2-2\delta} \cdot poly(d))$ $\square$

<u>Remark:</u> Assuming OVH, it is even hard to distinguish diam 2 from 3

$\Rightarrow$ No $(3/2 - \varepsilon)$-approx. in time $O((|E||V|)^{1-\delta})$ for diameter (under OVH)