Motivation

Idea: Study structure and hardness 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

```
0
```

(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 $APSP$ problem when

$n \approx n^2$ (dense graphs)

$O(n^3)$ 1962

$O(n^3/\log n)$ 1975

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

$O(n^{3/2} \sqrt{\log n})$ 2074

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

Deeper reason? For yes!

$n^{3-\varepsilon}$ alg for $APSP \iff n^{3-\varepsilon}$ alg for finding max. triangle

$\iff n^{3-\varepsilon}$ alg for finding min. triangle

$\iff n^{3-\varepsilon}$ alg for

10-15 more
Example: Longest common subsequence of two strings

A G C AT
G A C

$O(n^2)$ [Dhie et al '76]
$O(n^2 \log n)$ [Umasch/Patson '80]

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

Would be a major breakthrough in SAT solving

Comparison to "P vs NP":

Poly-time reducible

Max independent set

Poly-time reducible

Graph Coloring

NP-hard problems

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

$APSP \leftrightarrow SAT \leftrightarrow 3SAT \leftrightarrow 3SUM$

If in P: time complexity of reduction matters, polynomial.time is not enough!
Orthogonal Vector Problem

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

Task: Decide if there are $a \in A, b \in B$ such that $a \perp b$

\[ \langle a, b \rangle = 0 \iff \sum_{i=1}^{d} a_i b_i = 0 \iff \forall i \in d: a_i = 0 \text{ or } b_i = 0 \]

Example:

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

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

Algorithm: Time $O(n^2 d^*)$ - trivial

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

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

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 largest distance between any pair of vertices

\[ \max_{u,v \in V} \text{dist}_G(u, v) \]

(distance = length of the shortest path)

Single algorithm: single source shortest paths: \( O(m(\text{unweighted}) \cdot \log n) \) suggested

BFS

Dijkstra

all-pairs shortest paths: \( O(mn) \), \( O(mn \cdot \log n) \)

SSSP from every node

\( \Rightarrow \) sufficient to compute diameter in additional \( O(n^2) \) time

Theorem: Assuming OVH, there is \( O(1 \cdot m^{1.5}) \)-time algorithm (with constant \( s \geq 0 \)) working for computing the diameter of a graph.

Proof: Preprocessing step: make sure every vertex has a list of 1-ary nodes. Otherwise, output that there is no orthogonal pair

Reduction: Goal: diameter = 2 \( \iff \) no two orthogonal pairs

\[ \text{dist}(u, v) \leq 4^3 \text{ for all pairs } u, v \]

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

\[ \text{dist}(u, v) \leq 2 \text{ otherwise} \]

\( \Rightarrow \) only have to further analyze dist(0,1) or 1, 0 \in A, B

\[ \text{every } u \in A \text{ has some } k \text{ s.t. } 0[k] = 1 \]

\[ \text{if path } a = x, z, b \text{ then dist}(a, b) \geq 2 \]

Because no path edge are needed
If there is no orthogonal pair, then for every $a \in A$, $b \in B$
there is a $k$ s.t. $\alpha[k] = k'$ and $\beta[k] = 1$
\[ \Rightarrow \exists \, \text{path } \alpha \rightarrow k \rightarrow b \]
\[ \Rightarrow \text{dist} (\alpha, b) \leq 2 \]
\[ \Rightarrow \text{dist} (\alpha, \beta) = 2 \]

As for every other pair of nodes, dist $(\alpha, \beta) \leq 2$, we get
\[ \text{diam} (G) = 2 \]

If there is an orthogonal pair, then there are $A \in A$, $B \in B$, and
$\alpha \rightarrow \beta$, i.e., whenever $\alpha[k] = 1$, we have $\beta[k] = 0$.
No path from $a \rightarrow b$ has length $\leq 3$.
Enumerate all possibilities (simple paths):
- $\alpha \rightarrow \ell \rightarrow a' \rightarrow k$ for some $k$
- $\alpha \rightarrow \ell \rightarrow k \rightarrow r$ for some $k$
- $\alpha \rightarrow \ell \rightarrow k \rightarrow b'$ (if it exists) (for some $k$, $b'$)
- $\alpha \rightarrow k \rightarrow a' \rightarrow \ell$ (for some $k$, $a'$)
- $\alpha \rightarrow k \rightarrow a' \rightarrow k'$ (for some $k$, $k'$)
- $\alpha \rightarrow k \rightarrow r \rightarrow k'$ (for some $k$, $k'$)
- $\alpha \rightarrow k \rightarrow r \rightarrow b'$ (for some $k$, $b'$)
\[ \Rightarrow \text{dist} (\alpha, \beta) \geq 3 \]
We also know dist $(\alpha, \beta) \leq 3$ for all nodes $\alpha, \beta$
\[ \Rightarrow \text{diam} (G) = 3 \]

Thus, \[ \text{diam} (G) = 3 \text{ iff Orthogonal pair} \]

Some complexity: size of $G$: \[ |V| = 2n + d + 2 = O(n + d) \]
\[ |E| \leq 2n^2 + 2k + 2nk = O(nk) \]

Sonic for constructing $G = O(nk)$

Suppose $O1 \left( \frac{\log d}{\sqrt{\log n}} \right)$ time alg for diameter
\[ n \cdot d \cdot (n + d)^{-\frac{1}{5}} \]
\[ \Rightarrow \exists \text{ alg for } O(V \text{ with running time } C(n, d) = O(n^{2.35} \log^d d)) \]

Remark: assuming $OVH$, it is even hard to distinguish $\alpha \rightarrow \beta$ from $3$
\[ \Rightarrow \exists \text{ alg for } O(V \text{ with running time } C(n, \frac{n}{3}) \text{ for } \alpha \rightarrow \beta \text{ from } 3) \]