Model of Computation (Implicit)

Sequential – if an operation (e.g., comparison) takes one time unit, n such operations take n time units
→ single processor hardware

Different hardware → different properties

Parallel – if an operation takes one time unit on one processor, n such independent operations take one time unit on n processors.

Think about decision problems – imagine a room full of people
SAT: ① Each person randomly assigns literals and checks whether it satisfies relations
② If satisfies, stand up.

→ Improvement: Instead of being random, I could assign them to ensure full coverage of all possibilities

→ Optimization Problem: Similar, but must find a way to compare costs (eg. tour lengths) from each participant

→ Parallel computers exist in many configurations
- Independent machines connected on a network (SOTI@home)
- Shared memory machines
↳ Could spend an entire semester on these architectures and alg. (breaking apart problem; communication; coordination)

There are, of course, radically different computer "architectures" with very different properties

→ quantum computers — make use of superpositions of states and quantum properties

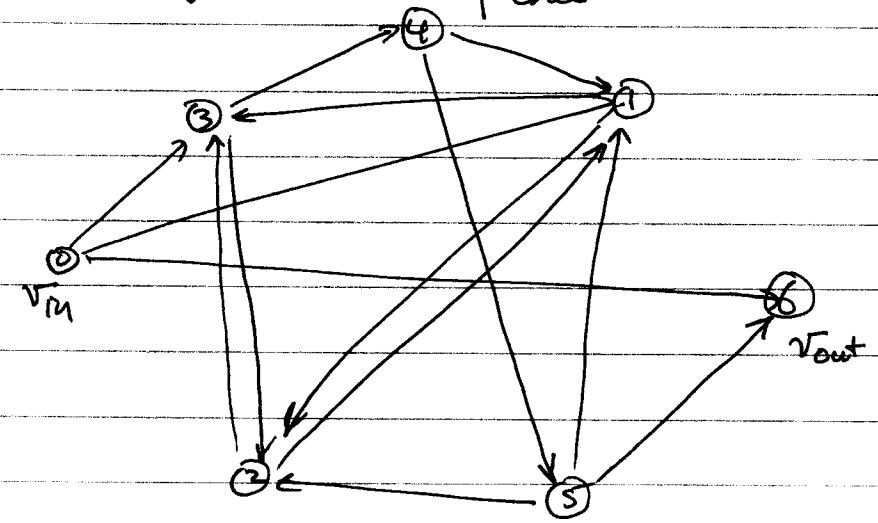
→ DNA computers — harnessing the power of molecular biology and the large numbers of molecules available in small volumes to do computation with molecules.

Hamiltonian Path Problem (Adleman, 1994)

Input: Directed graph $G(V, E)$ with two specific vertices v_{in} and v_{out}

Output: Decision (YES/NO) on whether there exists a path $v_{in} \to v_{out}$ that enters all other vertices exactly once

Ex:



14 edges

Yes: 1→2→3→4→5→6

This problem is NP-complete

Consider the following Algorithm

① Generate random paths through graph ← hopefully complete
② Keep only those beginning at v_{in} & ending at v_{out}
③ Keep only those paths entering $|V|$ vertices
④ Keep only those paths entering each vertex (at least once)
⑤ If any ~~graphs~~ path remain → YES; otherwise ⇒ NO

DNA Primer

a) There are four "bases" — A↔T C↔G (bind for complementation pairs)
b) A string of bases (synthesized as a contiguous polymer) binds tightly to its complement

— AAA CCC AAC CAC →
← TTT GGG TTG GTG —

c) Ligation glues adjacent strands into contiguous piece
d) gel electrophoresis separates by size
e) PCR amplification from desired endpoints
f) digest single stranded DNA

Actually do this experimentally (wet work)
DNA hybridization
Make random paths

① Each Vertex of graph represented by unique string (20-mer)

1: — 1_L | 1_R →
2: — 2_L | 2_R →
3: — 3_L | 3_R →
⋮

Each edge (i,j) represented as ← $\bar{i}_R$ | $\bar{j}_L$ —

eg. (1,2) ← $\bar{1}_R$ | $\bar{2}_L$ —

(← $\bar{2}_L$ | $\bar{3}_R$ | $\bar{3}_L$ —)

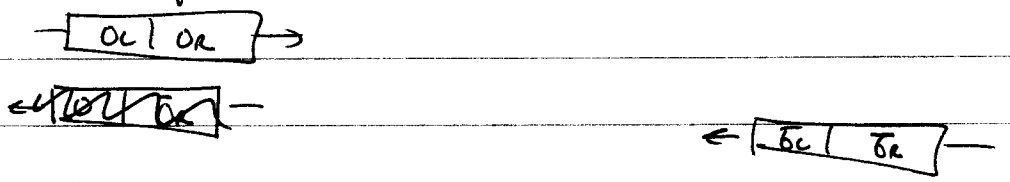
50 μm of DNA representing all edges and all vertices mixed (3×10^{13} copies/μl of each)

— 1_L | 1_R | 2_L | 2_R →
← T_R | 2_L ⋯ ⋯ ⋯ ⋯ ⋯ ;

⇒ Produce paths through the graph
Use ligation to chemically lock the strands

→ Keep only paths starting at $v_{in}=0$ & ending at $v_{out}=$

② PCR using

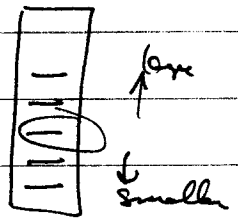


→ Amplifies (makes copies of) chains starting with 0 $v_{in}=0$ & ending with $v_{out}=6$. Others are diluted to negligible level

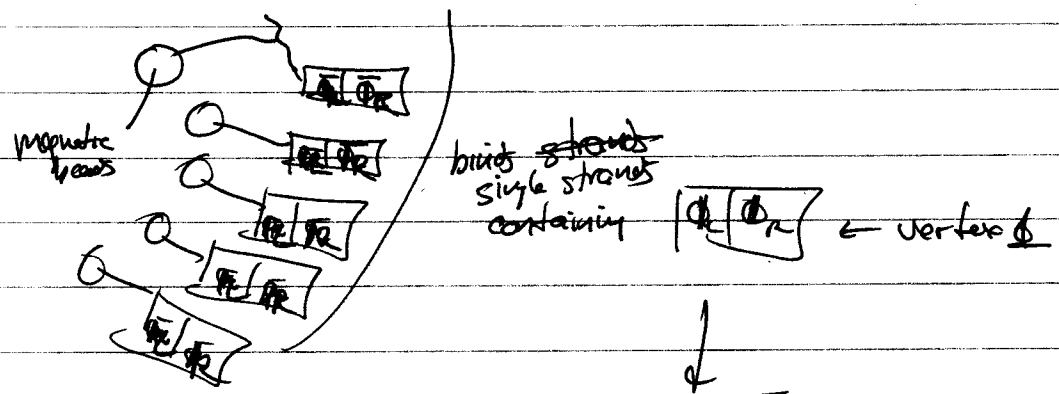
→ Pass through each vertex IV vertices

③ Separate DNA based on length (gel electrophoresis) and excise 140-bp lengths, purify.

(20×7)



④ Pass through each vertex at least once



⑤ Do any paths remain?

— Run on a gel and look for DNA → Reveals only one.

Summary:

one week of work

most time ("a whole day") on Step 4 (magnetic beads)

↳ time is linear in # of vertices ⇒ FULLY TIME, but on unequal architectures

Can actually read out the path,

though difficult to decode superposition of many solns.

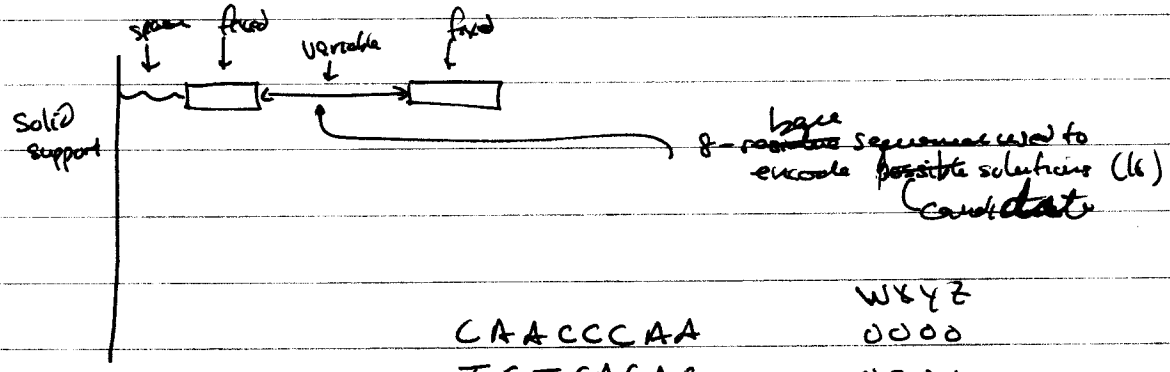
Problem — scaling up to interesting size problems (70 vertices) would require 10^{25} kg of material.

↑ ~ weight of Earth

SAT: (Liu et al., 2000)

$(W \vee x \vee y) \wedge (W \vee \bar{y} \vee z) \wedge (\bar{x} \vee y) \wedge (\bar{W} \vee \bar{y})$

4 binary vars $\Rightarrow 2^4 = 16$ candidate solns



	W&Y&Z
CAACCCAA	0000
TCTCAGAG	0001
⋮	⋮

① Synthesize all and attach to surface

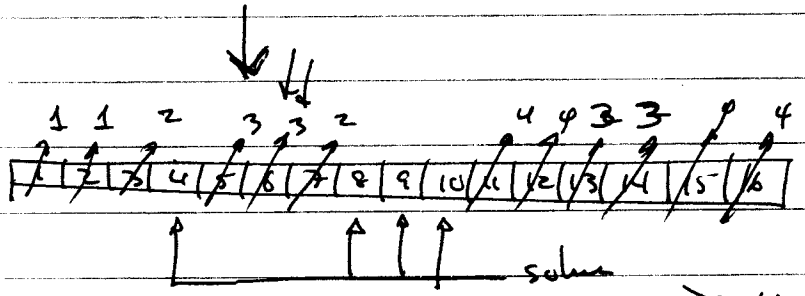
② Remove those incompatible with clause 1 $(W \vee x \vee y)$

$\Rightarrow$ 0000 & 0001 are the only incompatible

Protect Mark $\rightarrow$ $\Rightarrow$ So other 14 complements added, hybridized (to protect), digestion of single-stranded w/ exonuclease

Destroy $\rightarrow$

Protect Unmarked $\rightarrow$ $\Rightarrow$ Wash to de-hybridize



z: 0 x 1 0

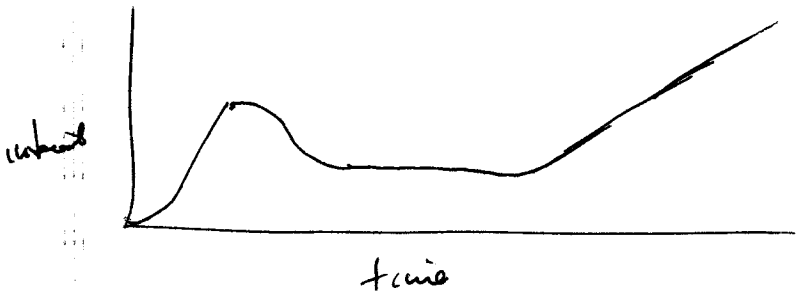
y: w 1 0 z

x: 1 x 1 z

4 solns: 0011
0111
1000
1001

→ Identify by PCR & hybridize to addressable array.

$\mathcal{O}$(# of mark & destroy operations grows polynomially in # of variables)



→ Holds tremendous potential; still has problems (scale-up).

→ Algorithms are very inefficient:

- Make all possible solutions and destroy members of search space and destroy themselves.
- Improvements that create only a small part of the space presumed to contain the solution would scale much better.