

Molecular Computation of Solutions to Combinatorial Problems

Leonard M. Adleman

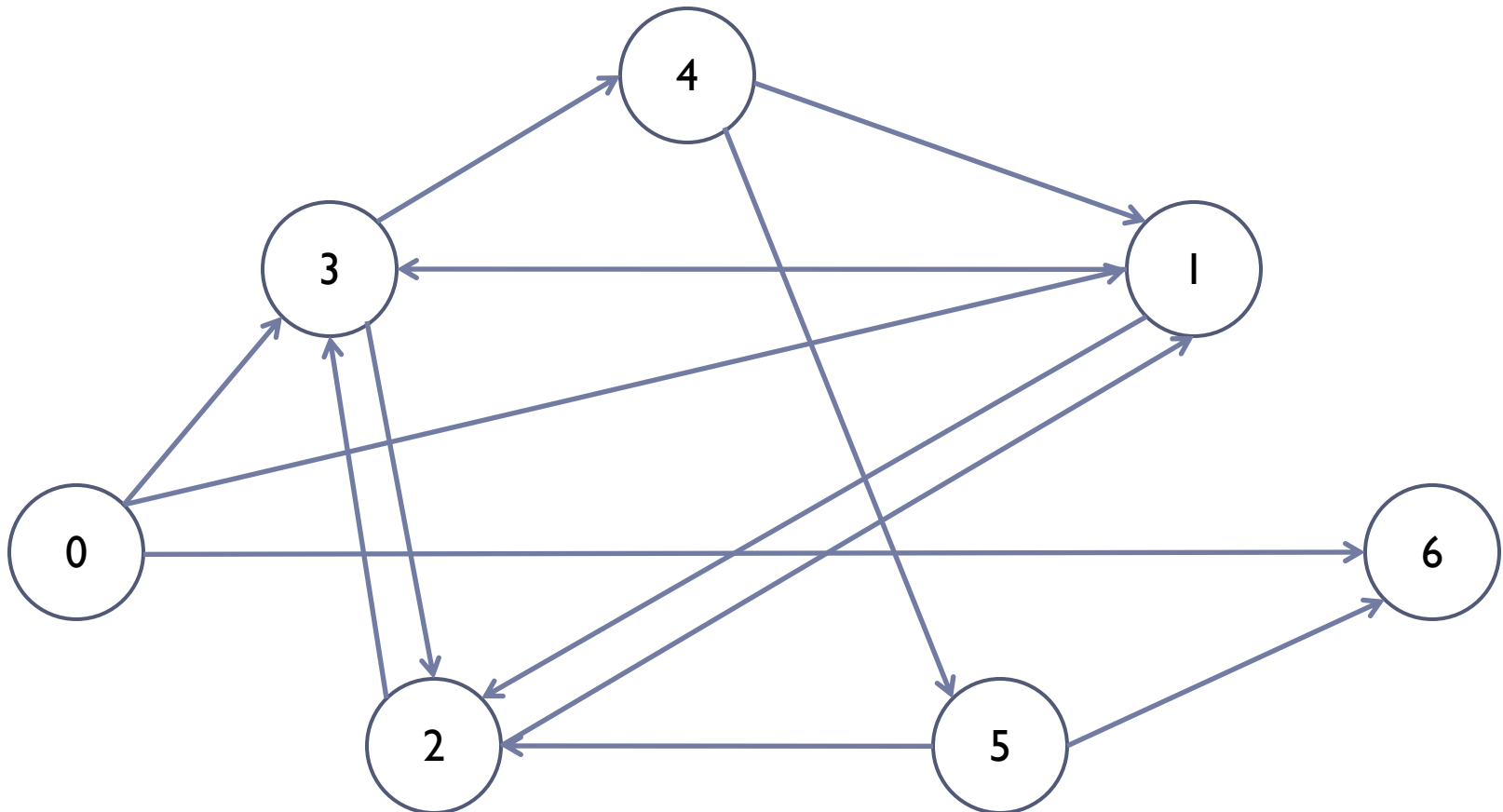
Jeff Hussmann, 3/4/09

Leonard M. Adleman

- ▶ Computer scientist
- ▶ The 'A' in RSA, the famous public key cryptography system
- ▶ Won the Turing award
 - ▶ Roughly equivalent to the Nobel prize in computer science

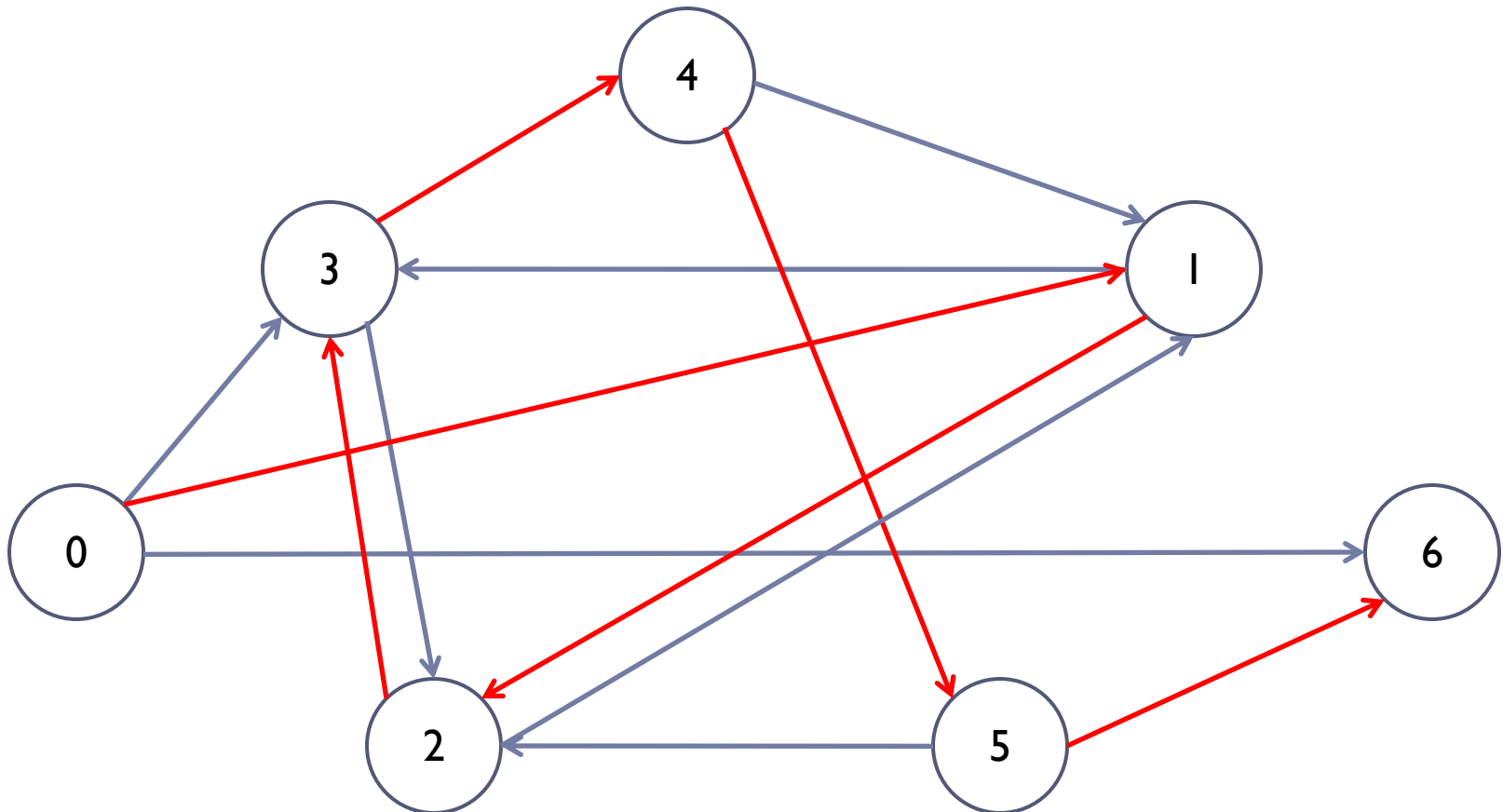
Hamiltonian Path Problem

Given a directed graph and designated starting vertex S and ending vertex E , does there exist a path from S to E entering every other vertex exactly once?



Hamiltonian Path Problem

If we take S to be vertex 0 and E to be vertex 6, the answer is yes.



Solving the Hamiltonian path problem

- ▶ All known conventional algorithms scale exponentially with the size of the graph (in fact, NP-complete)
 - ▶ As graph size increases, problem quickly becomes computationally intractable
- ▶ An unconventional (nondeterministic) algorithm:
 - ▶ **Step 1:** Generate random paths through the graph.
 - ▶ **Step 2:** Keep only those paths that begin at S and end at E .
 - ▶ **Step 3:** If the graph has n vertices, keep only those paths that enter exactly n vertices.
 - ▶ **Step 4:** Keep only those paths that enter all of the vertices of the graph at least once.
 - ▶ **Step 5:** If any paths remain, a Hamiltonian path has been found.

Step 1: Generate random paths through the graph

- ▶ Assign each vertex i in the graph a random 20 base pair sequence of nucleotides O_i
- ▶ Assign each edge from vertex i to vertex j the 20-base pair sequence $O_{i,j}$ consisting of the last 10 base pairs from O_i followed by the first 10 base pairs from O_j
- ▶ Mix many copies $O_{i,j}$ for each edge with the complements of O_i for each vertex and ligate

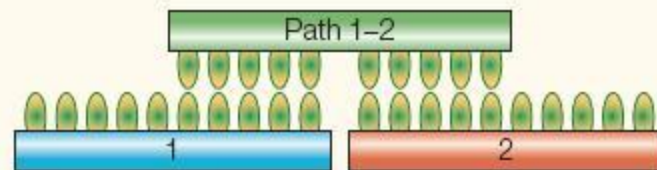
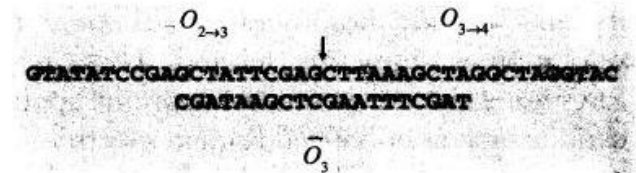
O_2 TATCGGATCGGTATATCCGA

O_3 GCTATTCGAGCTTAAAGCTA

O_4 GGCTAGGTACCAGCATGCTT

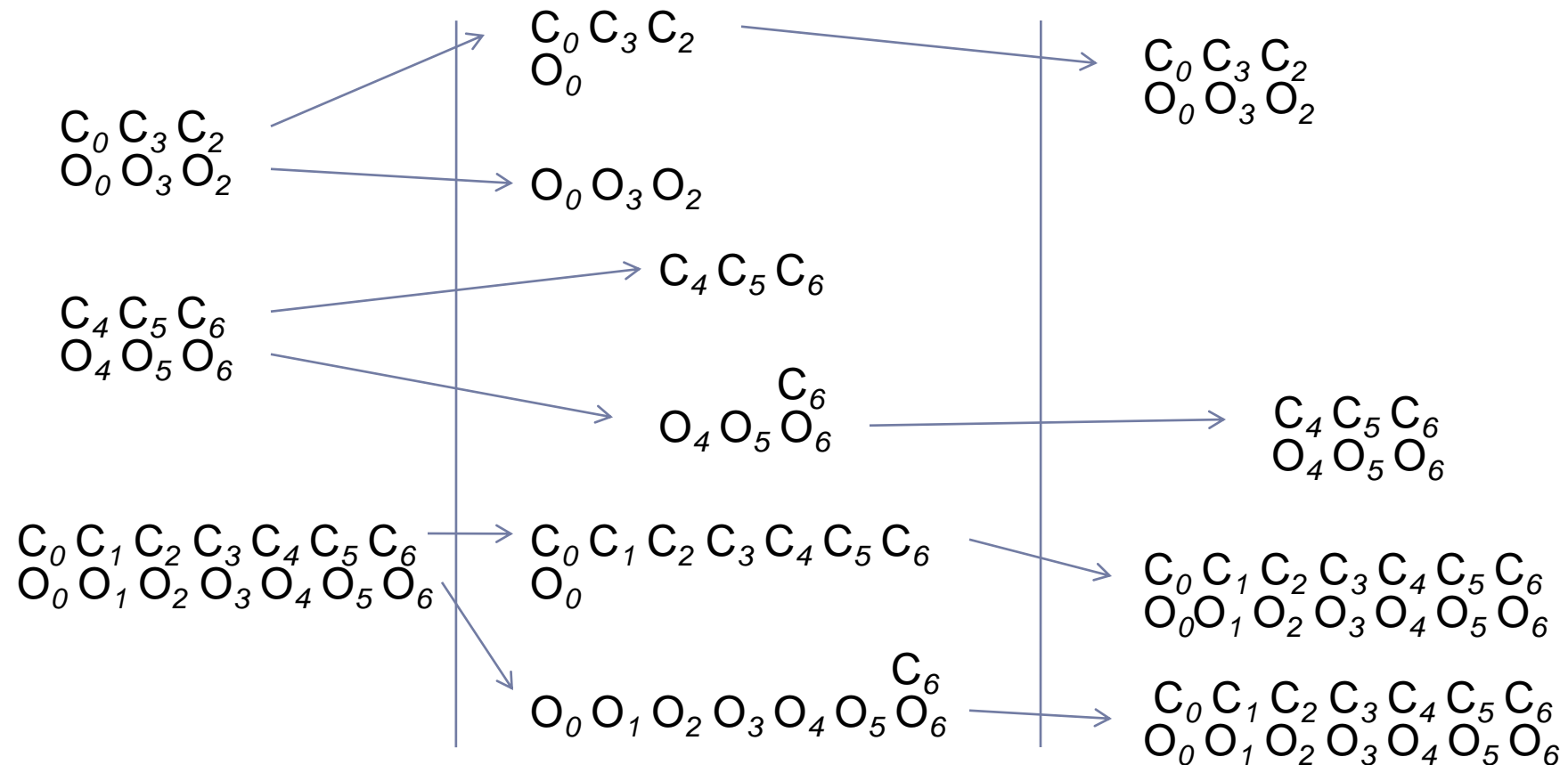
$O_{2 \rightarrow 3}$ GTATATCCGAGCTATTCGAG

$O_{3 \rightarrow 4}$ CTTAAAGCTAGGCTAGGTAC



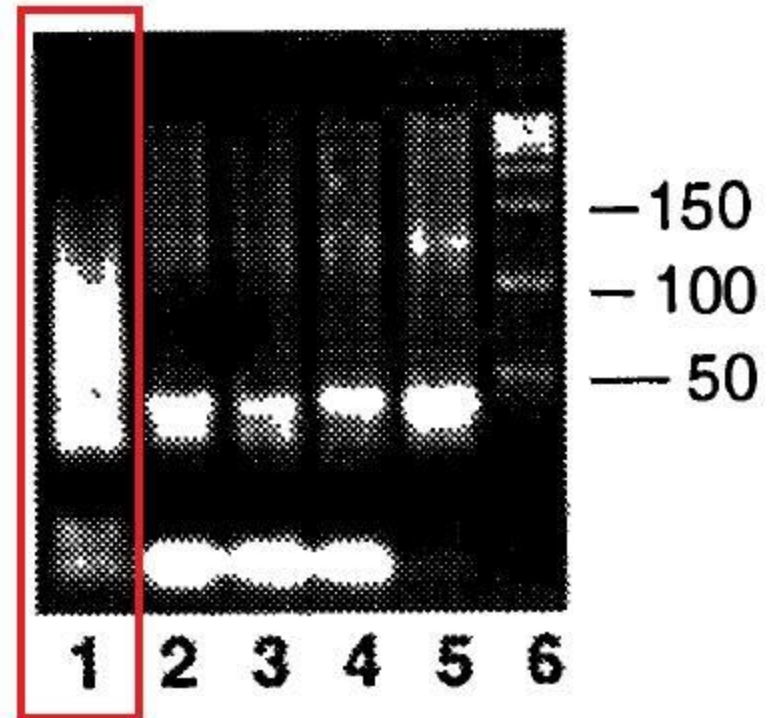
Step 2: Keep only those paths that begin at S and end at E

- Amplify the results of Step I via PCR, using O_0 and the complement of O_6 as primers



Step 3: Keep only those paths that enter exactly n vertices

- ▶ From an ensemble of DNA strands each consisting of 20 base pairs for each vertex in the path encoded, want to isolate those strands with 7 vertices in their path
- ▶ Differentiating DNA strands by length is simple – run on an agarose gel, extract the band which corresponds to sequences of 140 base pairs



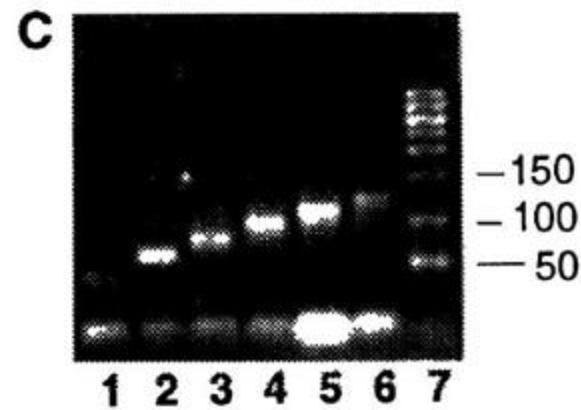
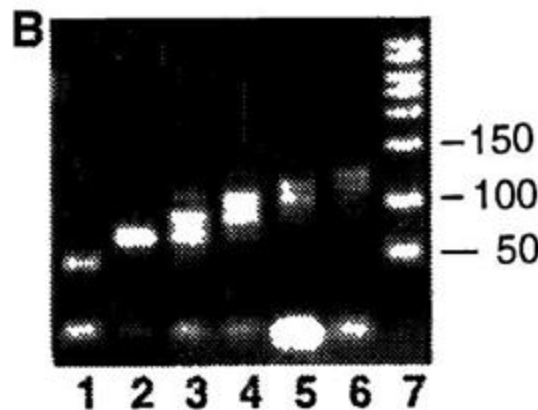
Step 4: Keep only those paths that enter all vertices of the graph at least once

- ▶ Since we are only working with paths of length 7 at this point, 'all at least once' is equivalent to 'all exactly once'
- ▶ 'Affinity purified with a biotin-avidin magnetic bead system'
 - ▶ A protein called biotin binds to a protein called avidin
 - ▶ Avidin binds to magnetic beads
 - ▶ For vertices $i = 1, \dots, 5$:
 - ▶ O_i is attached to biotin
 - ▶ Remaining DNA in single-stranded form is mixed with biotinylated O_i
 - ▶ Results washed over avidin-bound beads
 - Strands containing O_i are bound, strands not containing O_i are washed away

Step 5: If any paths remain, a Hamiltonian path has been found

- ▶ But what is it?
 - ▶ Could just sequence the resulting DNA
 - ▶ Instead, Adleman used a process called graduated PCR
 - ▶ Six PCR reactions are carried out and the results ran in 6 gel agarose columns
 - ▶ The i th column uses PCR primers of 0_0 and the complement of 0_i
 - The presence of a band at $20n$ base pairs in column i indicates that vertex i was visited in position n on the path

If this is done after step 3, we see superpositions of the paths (0, 1, 2, 3, 4, 5, 6), (0, 1, 3, 4, 5, 6), and (0, 3, 2, 3, 4, 5, 6).



After step 4, only the single Hamiltonian path remains.

Experimental Errata

- ▶ **Why 20 base pairs?**
 - ▶ A good primer length
 - ▶ Long enough to minimize ‘unintended bindings’
- ▶ **Steps aren’t perfect, only probabilistic**
 - ▶ PCR-based screening stages exponentially reduce concentration of undesired sequences, but don’t eliminate them
 - ▶ Note that one of the paths reflected in the post-step 3 gel image contains a path of length 6 that should have been eliminated
 - ▶ Errors in each step could be significant for larger problem sizes

So what?

▶ Processing power

- ▶ In 1994, the fastest supercomputers could carry out 10^{12} operations a second
- ▶ If concatenating (ligating) two DNA strands is considered an operation, the chemical reaction that carried out step 1 carried out approximately 10^{14} operations
 - ▶ Gain massive parallelization at the cost of precise control over operations
 - ▶ For certain problems, this may be a viable tradeoff

▶ Energy costs

- ▶ 1 joule of chemical energy could carry out 2×10^{19} operations of this kind
- ▶ Close to the (second law of thermodynamics mandated) limit of 34×10^{19} ops/Joule, much better than conventional supercomputers 10^9 ops/Joule

▶ Storage space

- ▶ Storing information in DNA strands in this way allows storage density of 1 bit per cubic nanometer
- ▶ Conventional media (in 1994) could only manage 1 bit per 10^{12} cubic nanometers

▶ The Conclusion

References

- ▶ “Molecular Computation of Solutions to Combinatorial Problems”, Leonard M. Adleman, Science, 11 November 1994.
- ▶ “The past, present and future of molecular computing”, Adam J. Ruben and Laura F. Landweber, Nature Reviews: Molecular Biology, October 2000.
- ▶ http://www.youtube.com/watch?v=_YgXcj4n-kQ (video about PCR)
- ▶ Wikipedia bio on Adleman