PATHS AND CYCLES IN GRAPHS

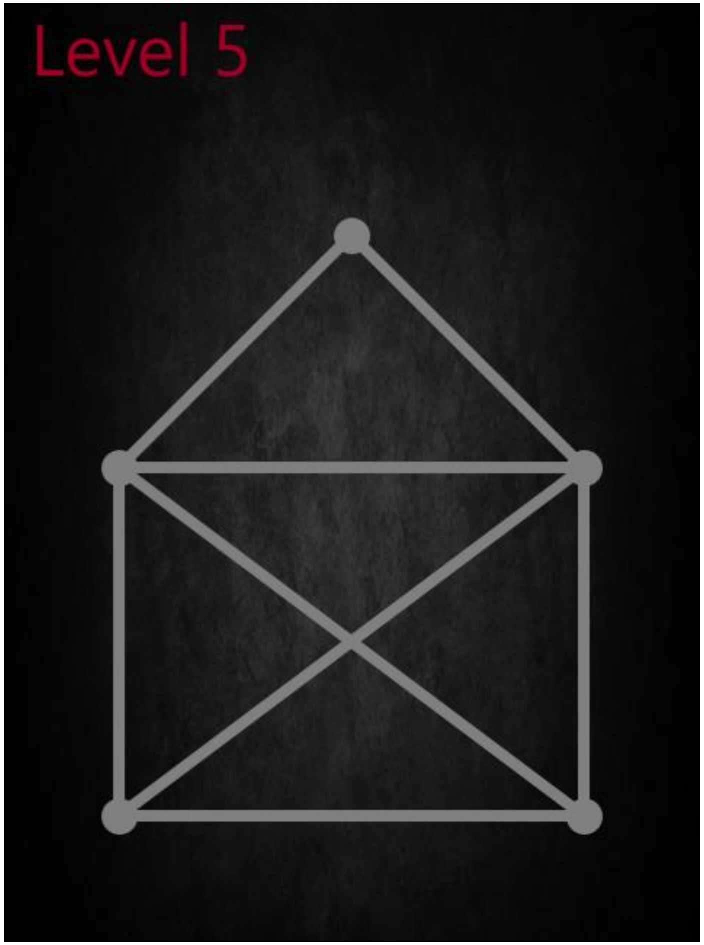


Figure Courtesy: One Touch Drawing App

- Can you draw this graph, without going over any edge twice, and without lifting your pencil?
  - *You need to find an Eulerian path on the graph!*

- An *Euler path* is a path that crosses every edge exactly once without repeating if it ends at the initial vertex, it's an Euler cycle

- A *Hamiltonian path* passes through each vertex (not edge), exactly once if it ends at the initial vertex, it's a *Hamiltonian cycle*

- In an Euler path, you might *pass through a vertex more than once*

- In a Hamiltonian path, you *may not pass though all edges*

WHAT IS THE CONNECTION BETWEEN HAMILTONIAN PATHS/EULERIAN PATHS AND BIOLOGY?

MORE RIDICULOUS STUFF: EXPLODING NEWSPAPERS!

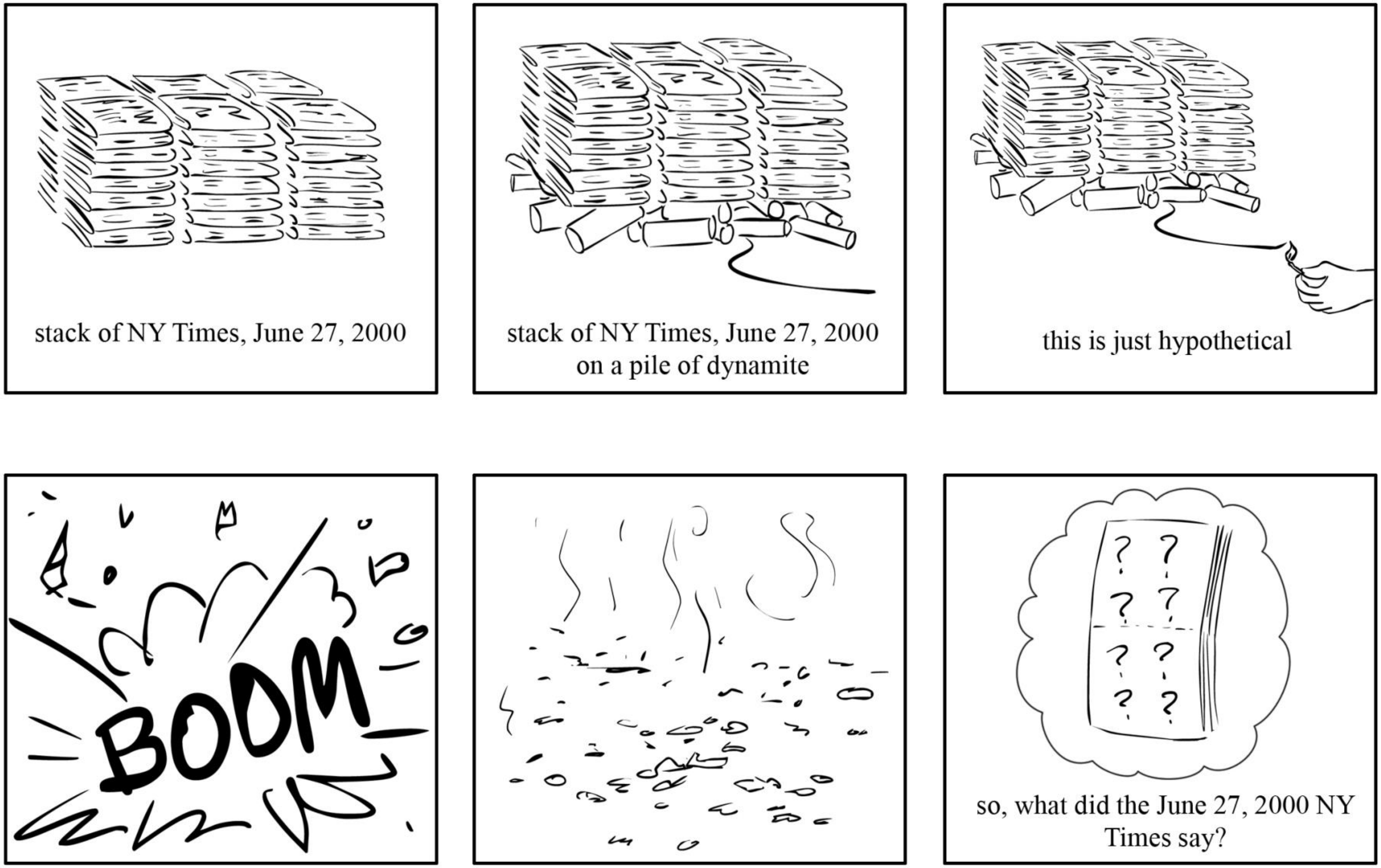

stack of NY Times, June 27, 2000
stack of NY Times, June 27, 2000 on a pile of dynamite
this is just hypothetical
BOOM
? ? ? ? ? ? ? ?
so, what did the June 27, 2000 NY Times say?

Fig 4.1 of Bioinformatics Algorithms by Compeau and Pevzner

THE GENOME ASSEMBLY PROBLEM

- Easier to sequence (generate reads), than to assemble them!
- Very large genomes can still not be assembled!

- What is the problem?
- **Given a (very) large number of [overlapping] reads, find a (smallest?) sequence that contains all of them**

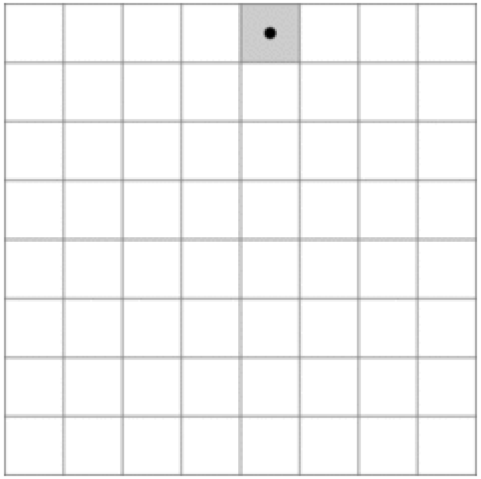
THE GENOME ASSEMBLY PROBLEM

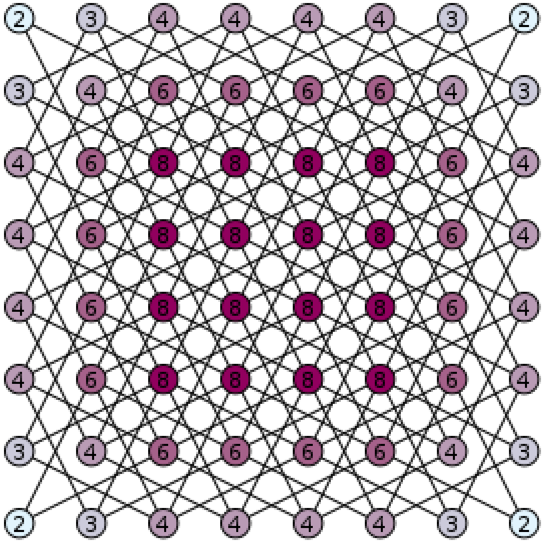
- *k*-mer composition of a string
- Composition$_3$(TATGGGGTGC):
  - {TAT, ATG, TGG, GGG, GGG, GGT, GTG, TGC}
  - {ATG, GGG, GGG, GGT, GTG, TAT, TGC, TGG} (sorted)

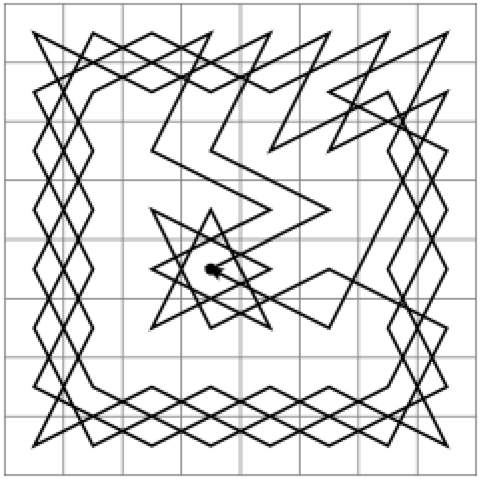
- Now, given the sorted set of *k*-mers, can you reconstruct the original string??
- Simple, create an overlap graph and then find a Hamiltonian!

ANOTHER EXAMPLE: KNIGHT'S TOUR

- Can a knight traverse the entire chessboard, visiting every square exactly once?
- Can you map this problem to a graph?
- What are the nodes, and what are the edges?







FINDING THE HAMILTONIAN

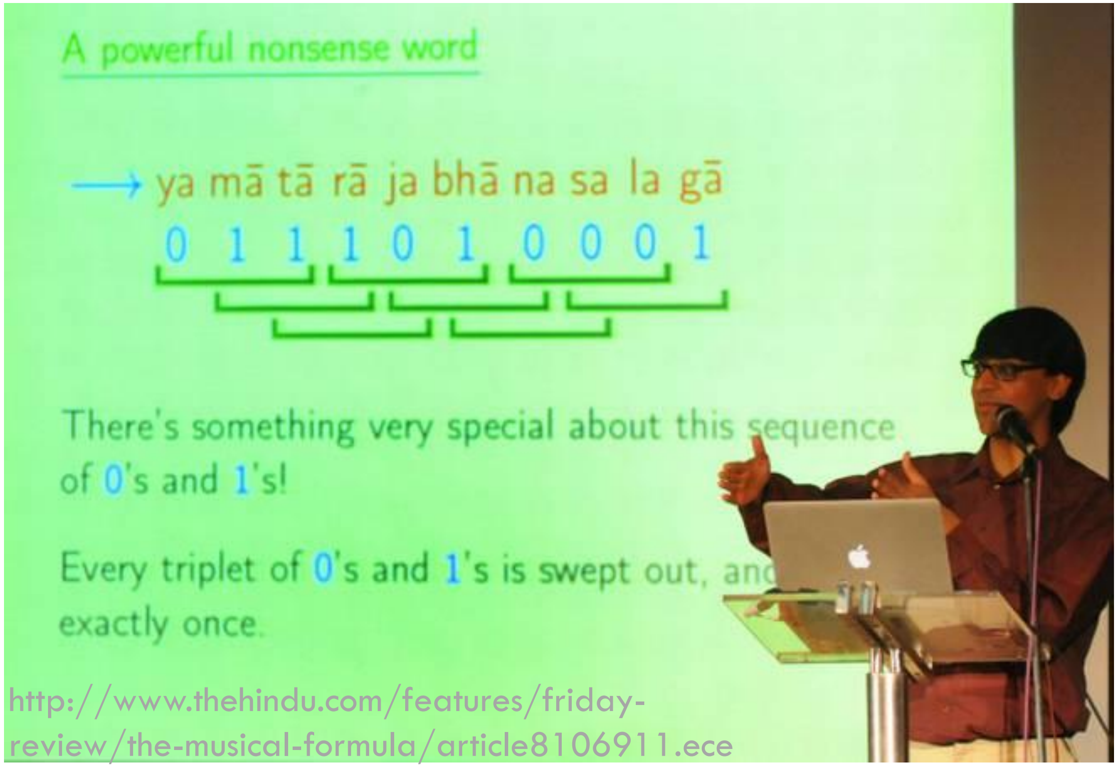
- Consider a set of *k*-mers:
  000, 001, 010, 011, 100, 101, 110, 111
- Can we find ~~the~~ a string that has the above as its *k*-mers?
- Represent every *k*-mer as a node in a graph
- Can we then find the shortest path that visits every node?
  - Obviously, it may not include all the edges
- Travelling Salesperson Problem (TSP): same as Hamiltonian cycle of least weight in a graph

ASIDE: यमाताराजभानसलगाम्

- What?
- yamātārājabhānasalagām
- Gibberish?

- Consider the reads 000, 001, 010, 011, 100, 101, 110 & 111
- What is the shortest string, that contains all these reads?
- यमाताराजभानसलगाम्!
- or 0111010001
  - Also see https://en.wikipedia.org/wiki/Sanskrit_prosody
- How do we find this string?

ASIDE: YAMĀTĀRĀJABHĀNASALAGĀM



A powerful nonsense word

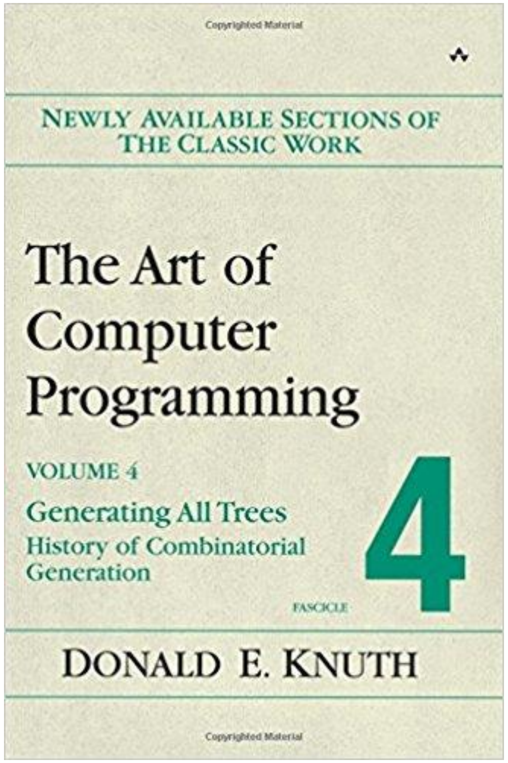
→ ya mā tā rā ja bhā na sa la gā

0 1 1 1 0 1 0 0 0 1

There's something very special about this sequence of 0's and 1's!

Every triplet of 0's and 1's is swept out, and exactly once.

http://www.thehindu.com/features/friday-review/the-musical-formula/article8106911.ece



Copyrighted Material

NEWLY AVAILABLE SECTIONS OF THE CLASSIC WORK

The Art of Computer Programming

VOLUME 4

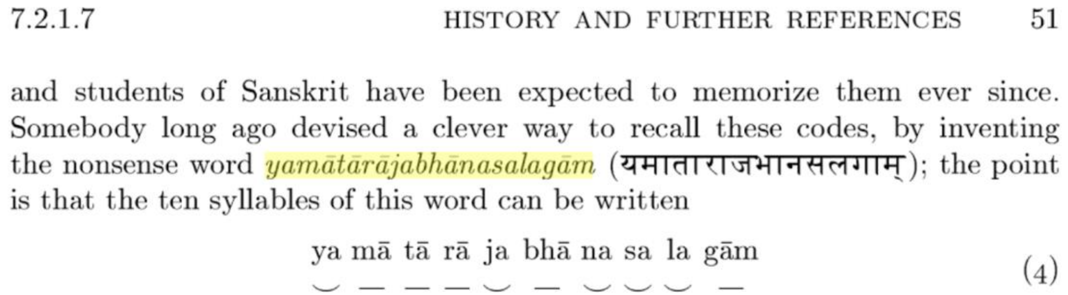
Generating All Trees

History of Combinatorial Generation

FASCICLE 4

DONALD E. KNUTH

Copyrighted Material



7.2.1.7 HISTORY AND FURTHER REFERENCES 51

and students of Sanskrit have been expected to memorize them ever since. Somebody long ago devised a clever way to recall these codes, by inventing the nonsense word *yamātārājabhānasalagām* (यमाताराजभानसलगाम्); the point is that the ten syllables of this word can be written

ya mā tā rā ja bhā na sa la gām (4)

⏑ — — — ⏑ — ⏑ ⏑ ⏑ —

MANY SUCH STRINGS EXIST!

- Many more challenges exist, in genome assembly!
- Have all bases been read?
- How much are the overlaps?
- Base calling errors?
- Which strand is this read from?
- What about repeats?!
- Read about de Bruijn **graphs**

0001011100

0001110100

0010111000

0011101000

0100011101

0101110001

0111000101

0111010001

1000101110

1000111010

1010001110

1011100010

1100010111

1101000111

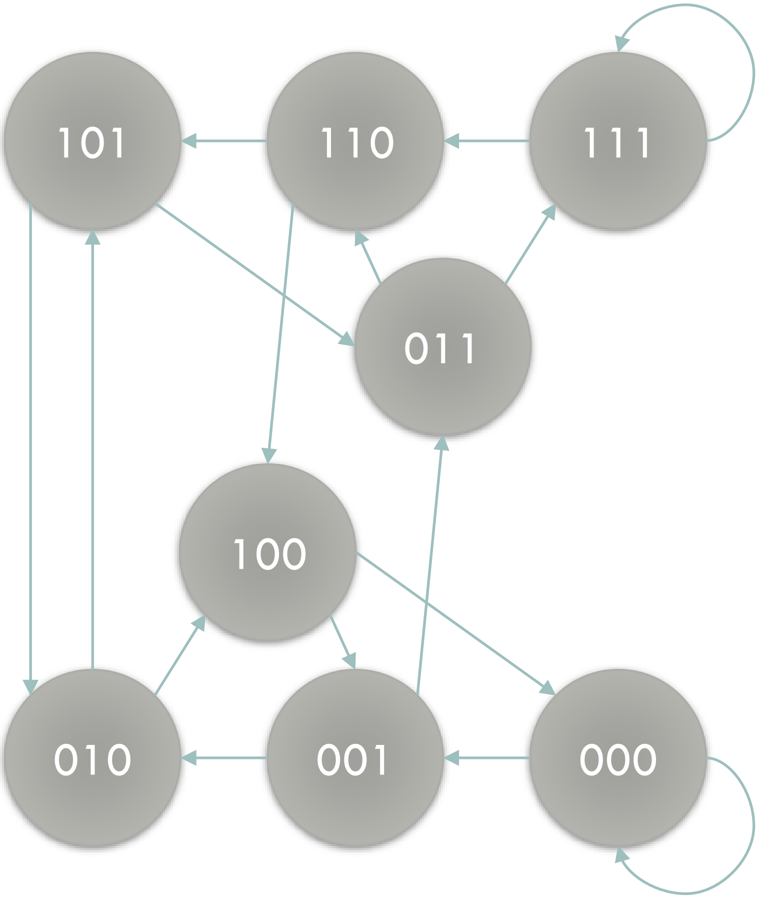
1110001011

1110100011

https://gist.github.com/karthikraman/ce98264aa6ade8f5802e

FINDING THE HAMILTONIAN IS HARD

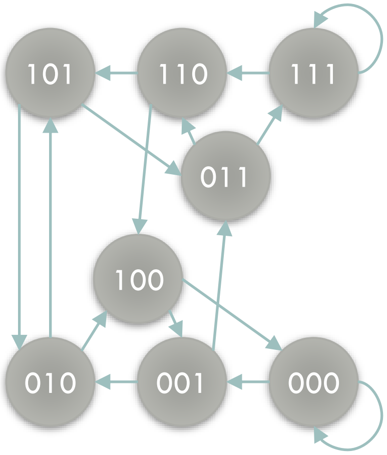
011
111
110
101
010
100
000
001

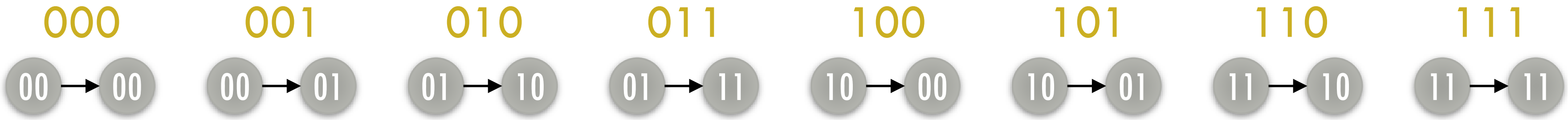


Genome: 0111010001

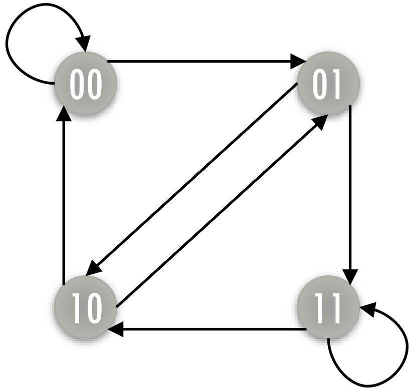
CAN WE FIND THE EULERIAN INSTEAD?

- Enter de Bruijn graphs
- de Bruijn (1946): Find a circular string containing every binary *k*-mer exactly once





Find the Eulerian in this graph:



Is this any easier? Why??

REMEMBER…

- We grossly simplified the genome assembly problem!
- In reality, so many more things need to be done!
  - Read pair sequencing: helps align sequences better
- Major assumptions
  - Reads are not error free
  - Perfect coverage of the genome by reads
- But basic ideas do not change!
- Pavel Pevzner: *"There's a saying in bioinformatics, to simplify the problem to the point it becomes ridiculous, solve this problem, i.e. develop the key idea for solving a ridiculous problem, and then overcome some ridiculous assumptions!"*

RAW READS & FASTQ

- File type: `fastq`
- Each sequence is described over 4 lines
  - Sequence id: begins with `@`
  - Read sequence
  - Additional information: begins with `+`
  - Quality score: lowest ! – highest ~ (ASCII characters!)

```
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
```

- No standard file extension
  - `.fq, .fastq, .sequence.txt, .fq.gz` (compressed)

- See http://rosalind.info/problems/tfsq/