De Bruijn Graphs

Ian Chen

Outline

De Bruijn Sequences

De novo Genome Assembly

Section 1

De Bruijn Sequences

Problem Statement

For $n \geq 1$, does there exist a circular sequence S that contains all n-length binary strings exactly once?

Examples

For $n = 1$, we uniquely have

10

For $n = 2$, we uniquely have

1100

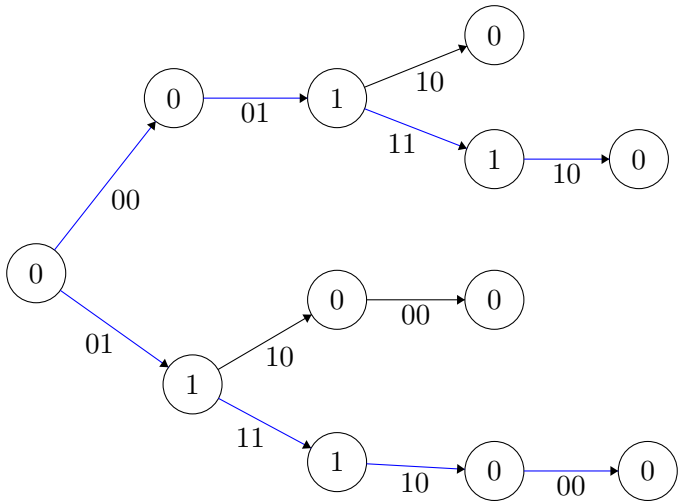
For $n = 3$, we have

11100010 11101000

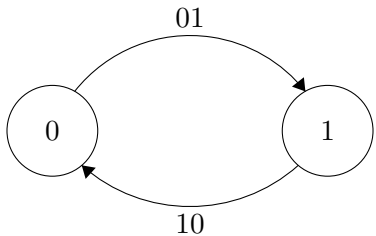
Observations

- S must have length exactly 2^n
- Every $(n-1)$-length substring occurs exactly twice
- The first n bits are arbitrary

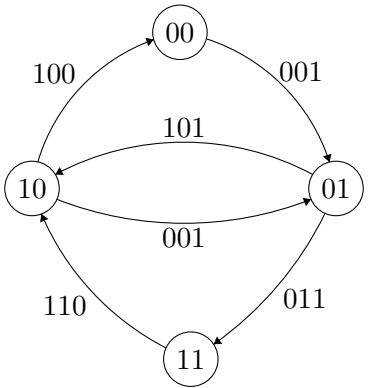
A Graph Representation



A Better Graph Representation



A Better Graph Representation



De Bruijn Graph

Let $n \geq 1$.

$$V = \mathbb{F}_2^{n-1}$$

$$u \to v \text{ if } \textit{suffix}(u) = \textit{prefix}(v)$$

What graph problem are we solving?

Eulerian Circuit

Definition (Circuit)

A *circuit* is a closed walk that uses an edge at most once.

Definition (Eulerian Circuit)

A circuit is *Eulerian* if it uses all edges exactly once.

Eulerian Graphs

We say a (simple) (di)graph is Eulerian if it has an Eulerian circuit.

Theorem (Eulerian Graphs)

G is Eulerian if $\text{indeg}(v) = \text{outdeg}(v)$ *for all* $v \in V(G)$.

Eulerian Graphs Sufficiency

FindEulerianCircuit(G):
  $T \leftarrow$ maximal trail in G
  $G' \leftarrow G \setminus T$
  $C \leftarrow$ FindEulerianCircuit(G')
  return $T \cup C$

Eulerian Graphs Sufficiency

Proof.

Suppose G is balanced. Then, T must be a circuit. G' must be balanced. Then, $T \cup C$ is Eulerian, by induction. $\square$

Claim

The above algorithm runs in $O(|E|)$ time.

De Bruijn Sequences

Claim

For all n, De Bruijn sequences exist.

Proof.

Consider G, the De Bruijn graph. For all $v \in V(G)$,

$$\mathit{indeg}(v) = \mathit{outdeg}(v) = 2$$

Thus, G is Eulerian. $\square$

Additional Results

- There are $2^{2^{n-1}-n}$ such circular sequences
- Considering not-circular, there are $2^{2^{n-1}}$ sequences
- There is a bijection between pairs of De Bruijn sequences, and all binary 2^n sequences.

Section 2

De novo Genome Assembly

Sequencing Is Hard

- First generation (Sanger) sequencing: Sorting Based
- Next generation sequencing: Synthesis Based

We get short k-mers, rather than long sequences.

Problem Statement

There is a model string T. Given all k-mers, estimate T.

Law of Assembly

If $\mathit{suffix}(A) = \mathit{prefix}(B)$, they might overlap.

$$ACG \cup CGT \implies ACGT$$

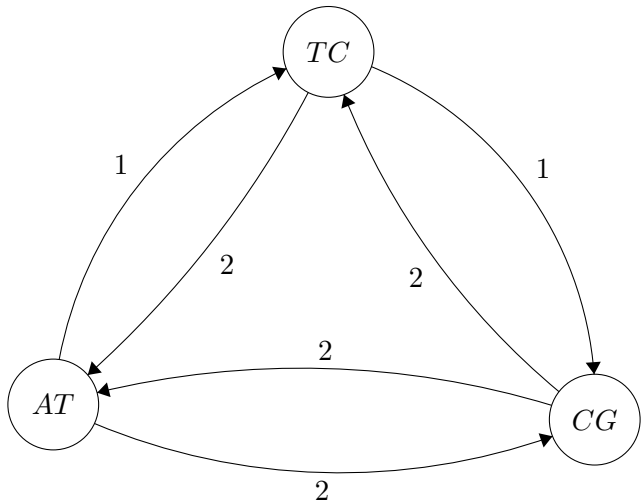
Overlap Graphs (SCS)

Form a connected graph $G = K_n$.

$$w(A, B) = \mathit{suffix}(A) = \mathit{prefix}(B)$$

We want to find the shortest common superstring.

Overlap Graphs (SCS)



Overlap Graphs (SCS)

This is the *Traveling Salesman Problem*. It is *NP-Hard*

We may approximate with greedy Nearest Neighbor. This is a $\log n$ approximation.

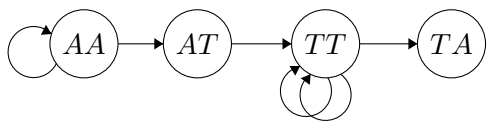
Additional Remarks

- Determining Overlaps: bloom filters
- Tandem Repeats: $AAAAA$

De Bruijn Graphs

Break all k-mers into 2 $(k-1)$-mers. Create De Bruijn graph.

If the genome is $AAATTTA$, for $k = 3$,

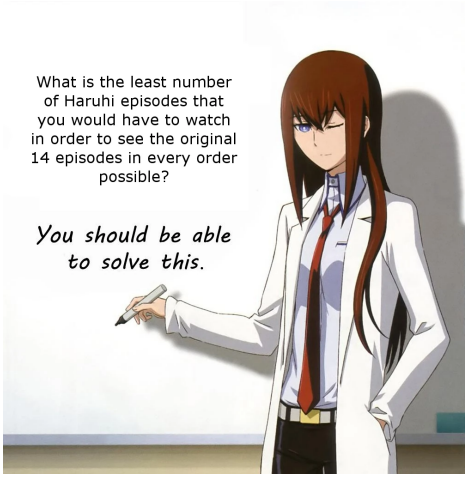


We want to find an *Eulerian Trail*.

Additional Remarks

- Unequal coverage, repeats
- Error correction,
- Bubbles, islands

Brainteaser



WAGA WAGA

— Sariel Har-Peled (2024)

Bibliography I

- Ben Langmea (JHU), https://www.langmead-lab.org/teaching.html
- Lionel Levine (MIT), https://pi.math.cornell.edu/~levine/18.312/