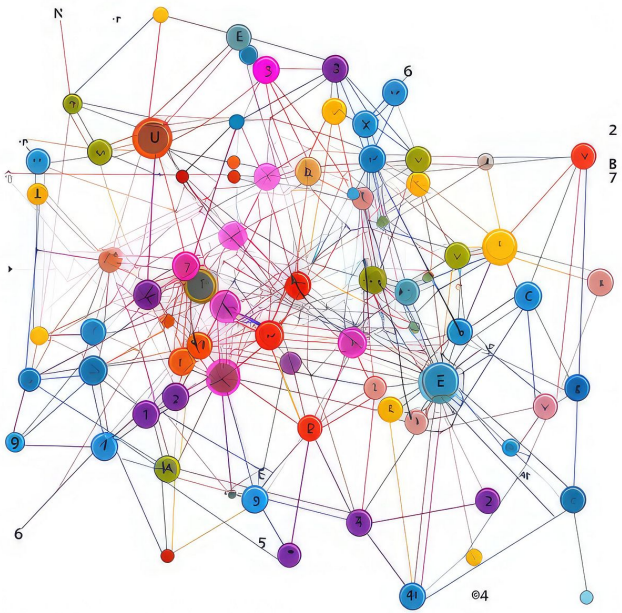
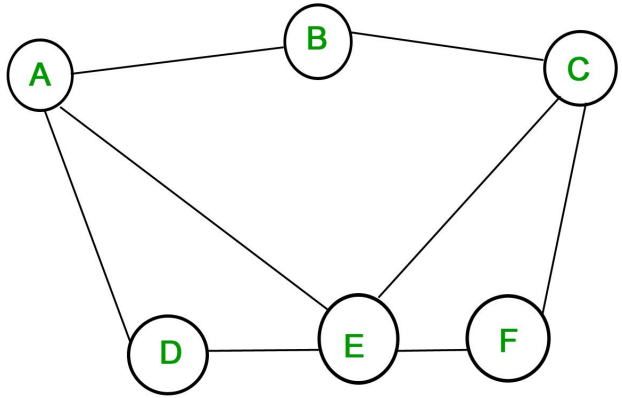
Montaje de secuencias

Grafos o gráficas





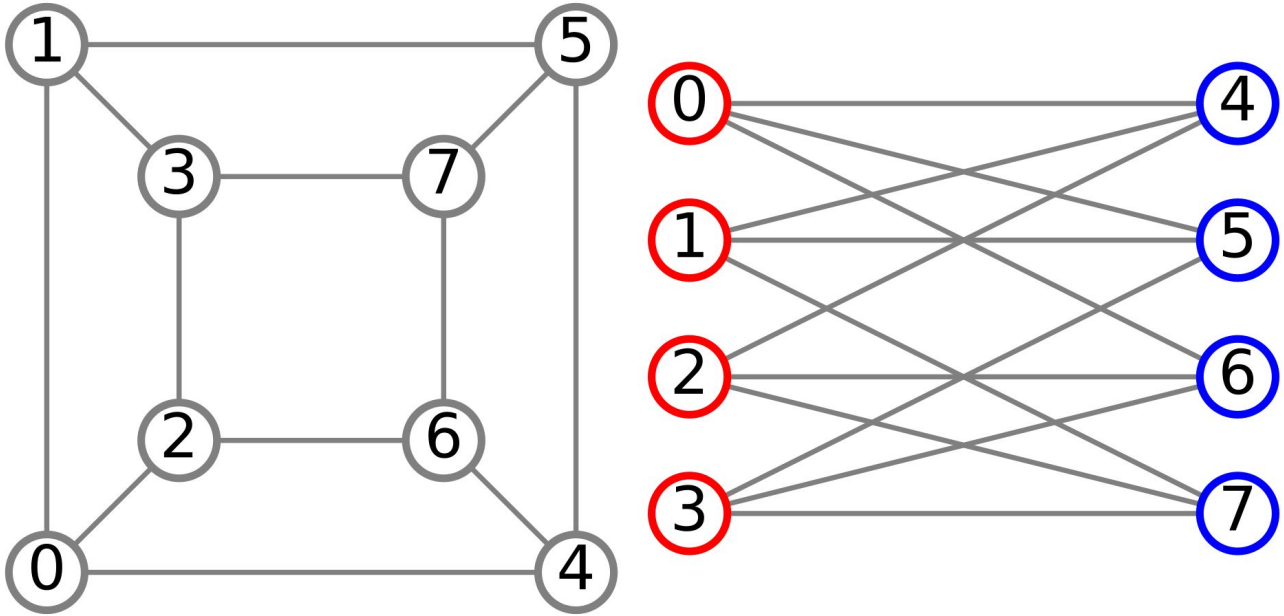
¿Qué es un grafo?

Un **grafo no dirigido** *G(V,E)* es una pareja ordenada en donde V es un conjunto no vacío de vértices y E es el conjunto de aristas, el cual consta de pares **no ordenados** de vértices, como {x,y}.

Un **grafo dirigido** *G(V,E)* es una pareja ordenada en donde V es un conjunto no vacío de vértices y E es el conjunto de aristas, el cual consta de pares **ordenados** de vértices, como (x,y).

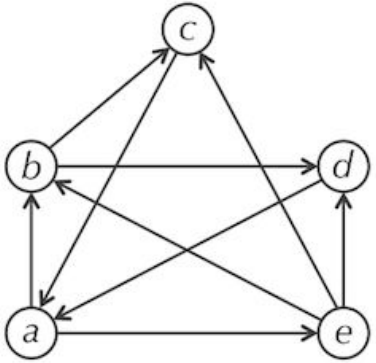
Puentes de Königsberg

Isomorfismo de grafos



Representaciones de grafos

Graph



Adjacency Matrix

	a	*b*	*c*	*d*	*e*
a	0	1	0	0	1
b	0	0	1	1	0
c	1	0	0	0	0
d	1	0	0	0	0
e	0	1	1	1	0

Adjacency List

a is adjacent to *b* and *e*
b is adjacent to *c* and *d*
c is adjacent to *a*
d is adjacent to *a*
e is adjacent to *b*, *c*, and *d*

Secuenciación de ADN

Un investigador toma una pequeña muestra de tejido que contiene millones de celulas con ADN idéntico, se utilizan métodos bioquímicos para quebrar el ADN en fragmentos y luego se secuencian estos fragmentos para generar los “reads”.

La tarea de utilizar estos fragmentos para reconstruir la cadena original se le conoce como montaje de secuencias.

Multiple identical copies of a genome

Shatter the genome into reads

Sequence the reads

Assemble the genome using overlapping reads

AGAATATCA
GAGAATATC
TGAGAATAT
...TGAGAATATCA...

Algunos retos

- No sabemos a priori que hebra estamos analizando en cada read
- Las máquinas modernas de secuenciación no son perfectas
- Algunas regiones del genoma no van a ser cubiertas por algún read

En este caso, vamos a asumir que no existen errores y que los métodos modernos alcanzan a secuenciar todo el genoma.

Composición de k-meros

Dada una cadena texto, la composición de k-meros es la colección de todas las subcadenas k-meros en dicha cadena.

$$Composition_3(\text{TATGGGGTGC}) = [\text{ATG}, \text{GGG}, \text{GGG}, \text{GGT}, \text{GTG}, \text{TAT}, \text{TGC}, \text{TGG}].$$

Reconstrucción de una cadena

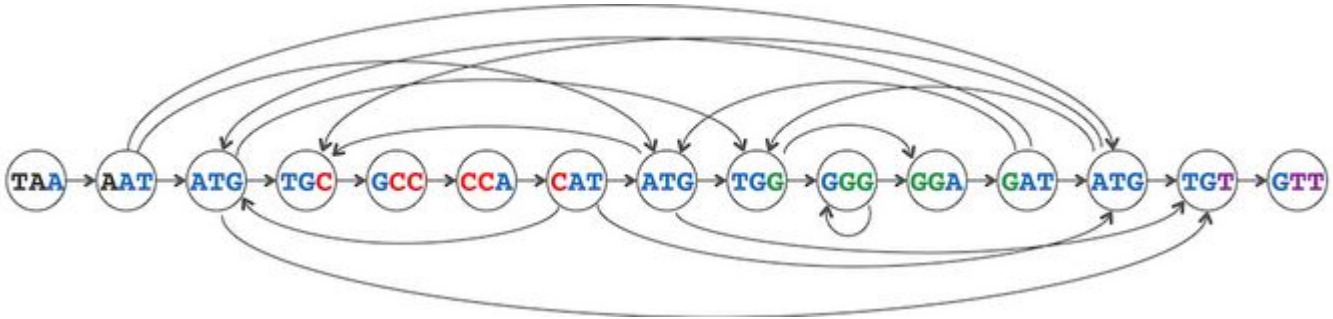
Una reconstrucción más larga

AAT ATG ATG ATG CAT CCA GAT GCC GGA GGG GTT TAA TGC TGG TGT

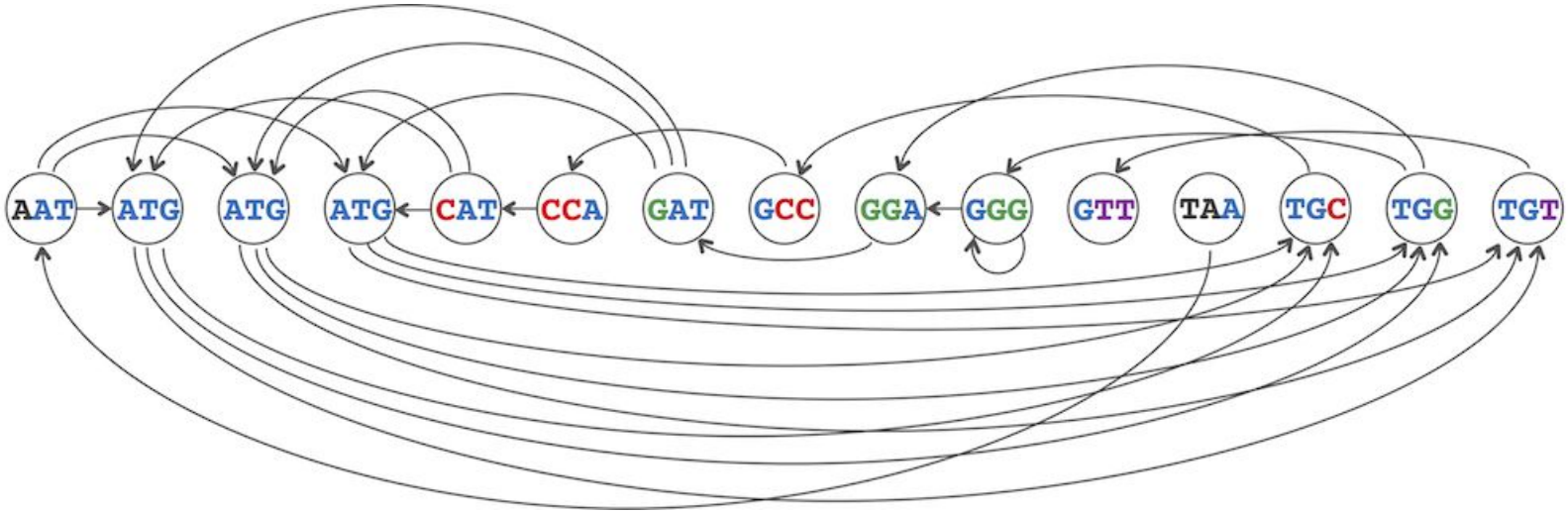
Utilizamos grafos

TAA
AAT
ATG
TGC
GCC
CCA
CAT
ATG
TGG
GGG
GGA
GAT
ATG
TGT
GTT
TAATGCCATGGGATGTT

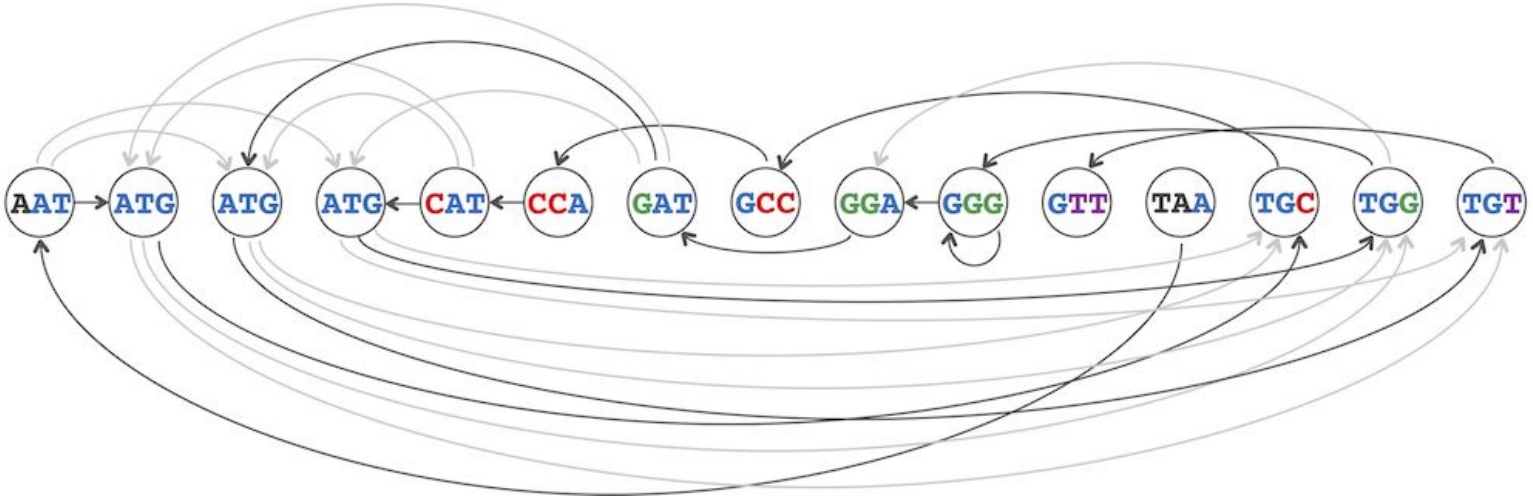
Grafo dirigido



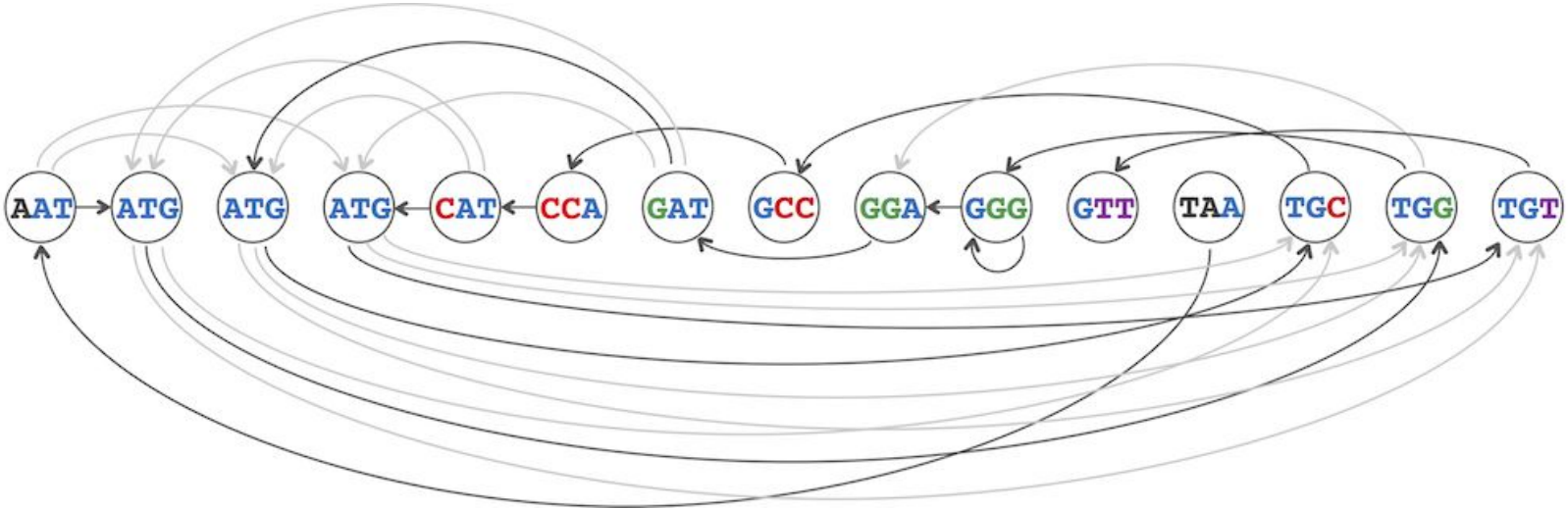
Si desordenamos un poco…



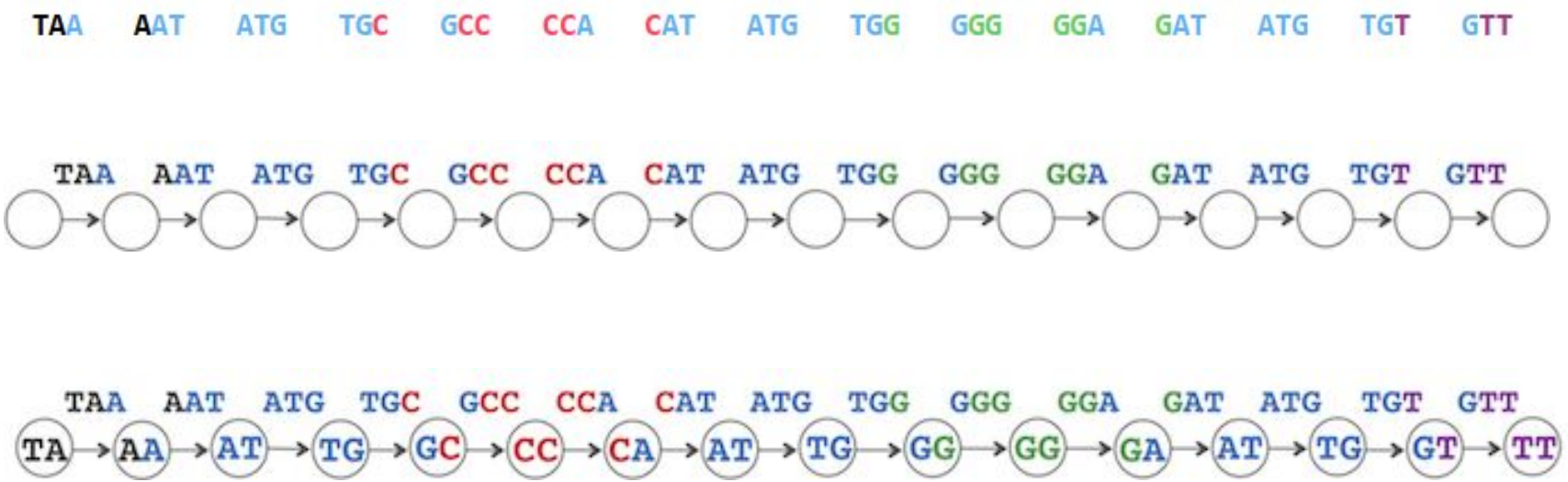
Si desordenamos un poco…



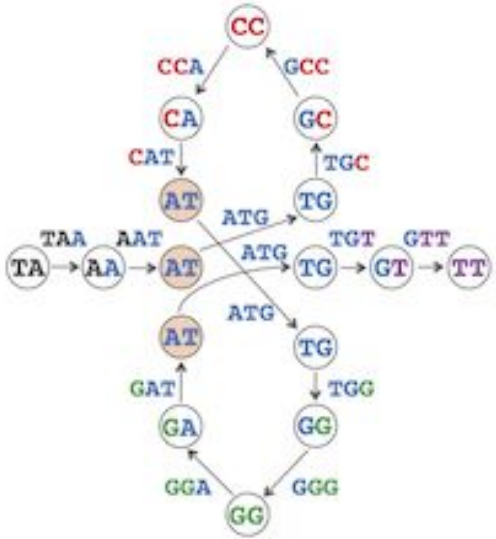
Caminos Hamiltonianos

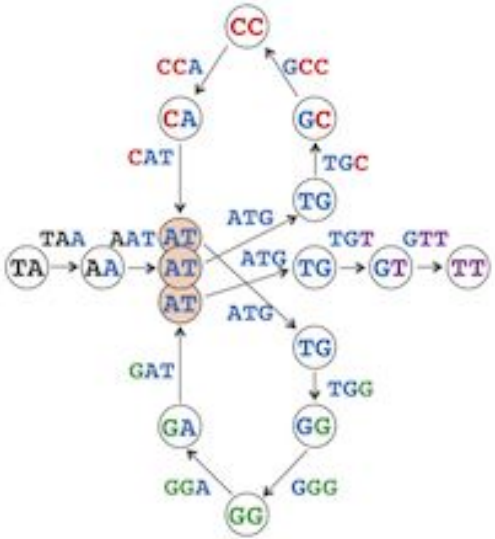


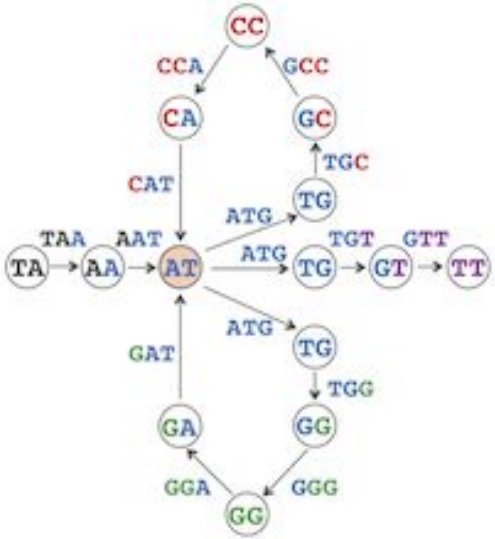
Otra forma de construir grafos



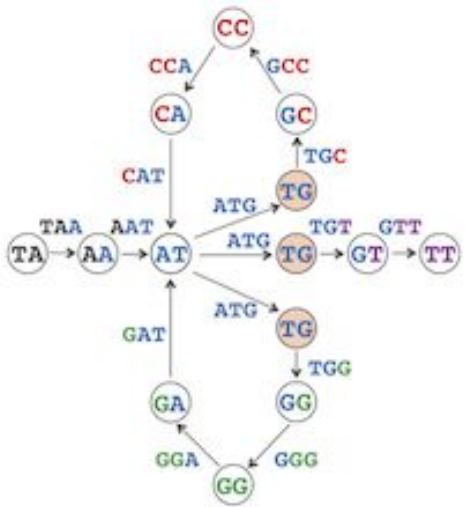
Transformamos el grafo

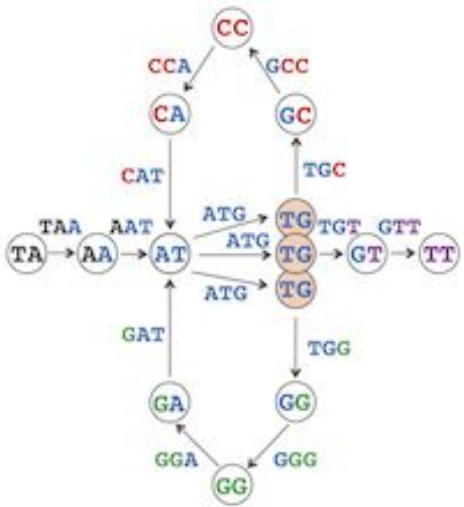


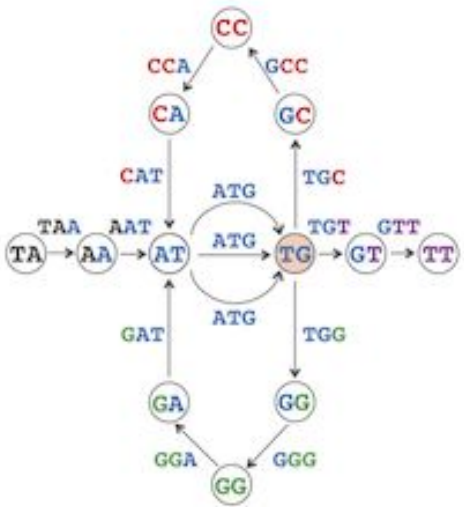




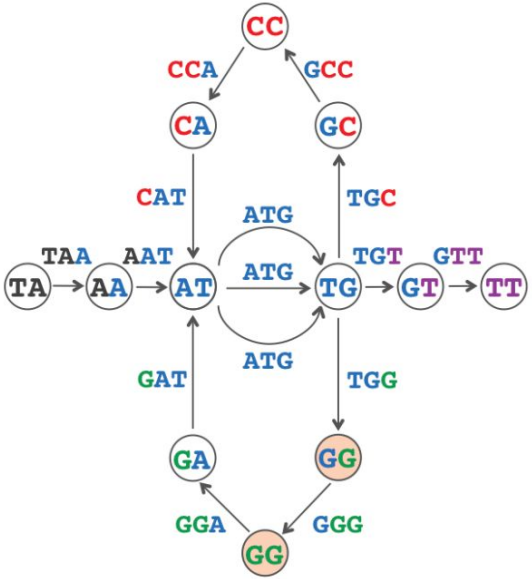
Transformamos el grafo

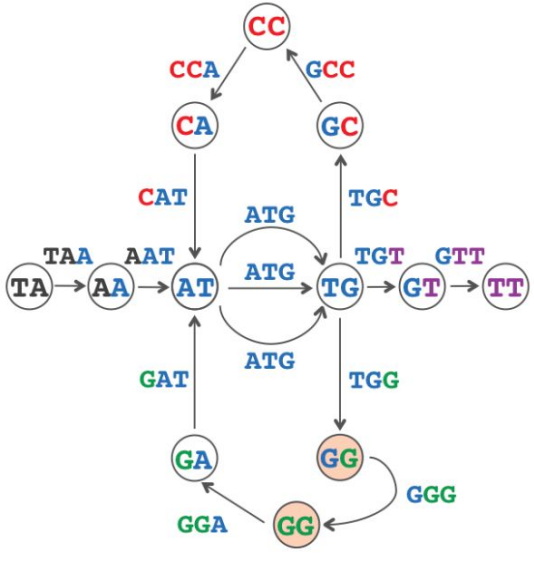


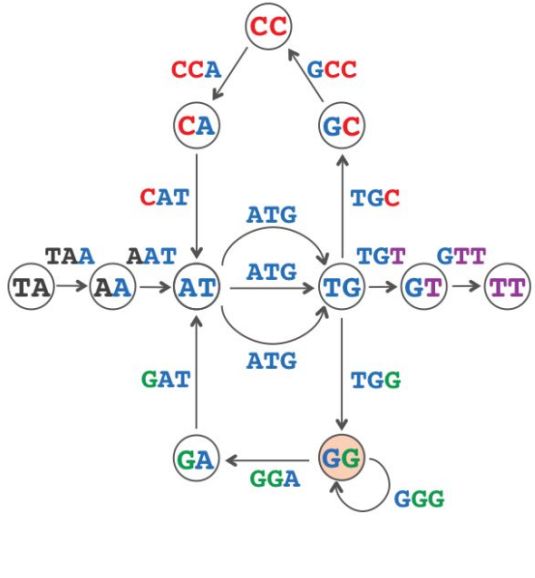




Transformamos el grafo

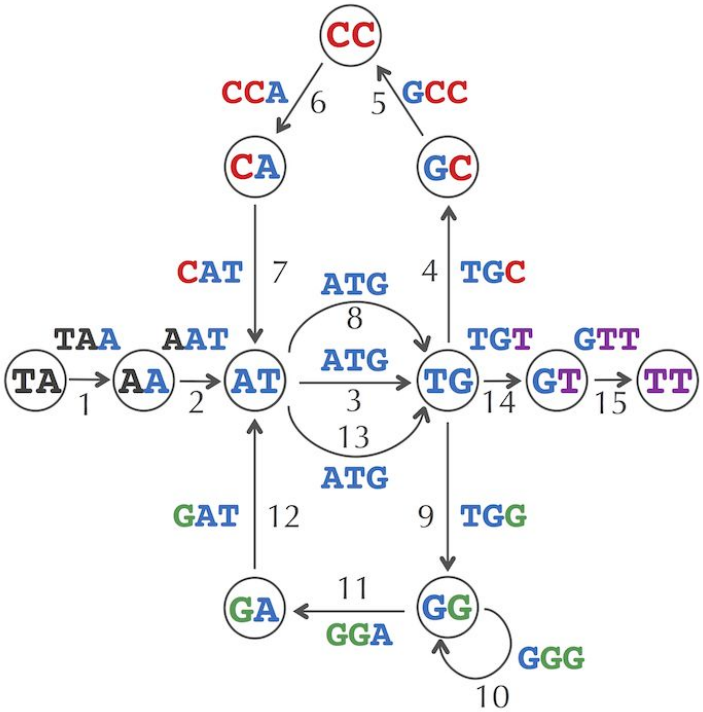




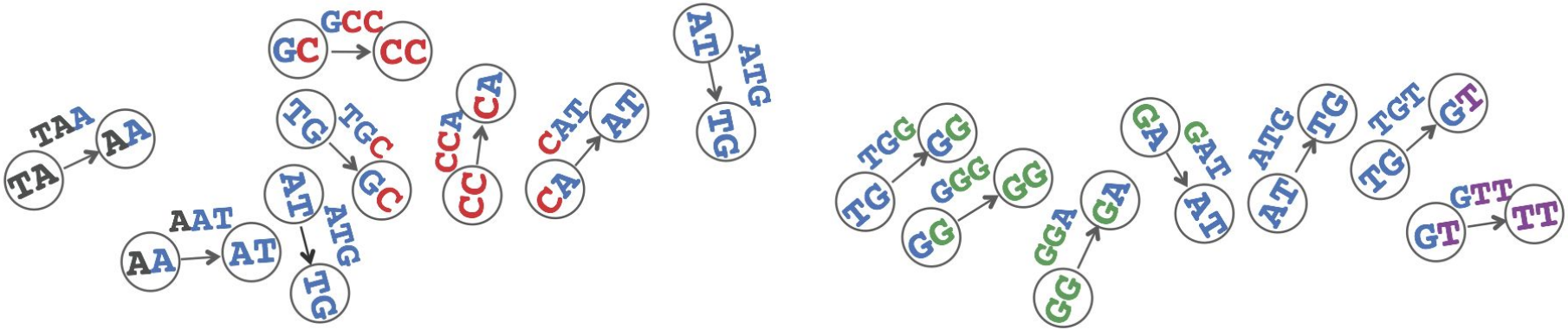


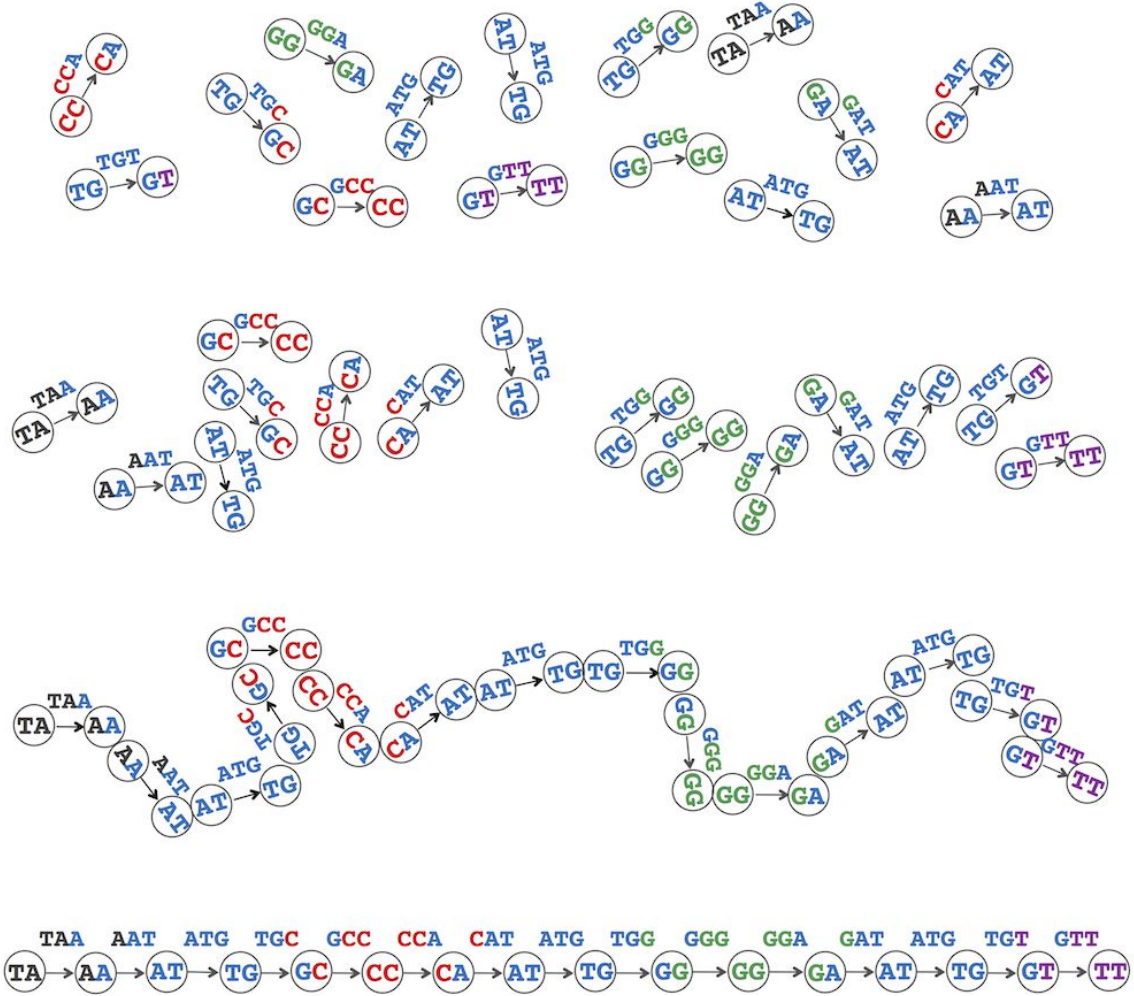
Caminos eulerianos

“Un camino o recorrido por un grafo que usa todas las aristas solo una vez”



Otra forma de construir Grafos de De Bruijn




TAA AAT ATG TGC GCC CCA CAT ATG TGG GGG GGA GAT ATG TGT GTT
TA AA AT TG GC CC CA AT TG GG GG GA AT TG GT TT

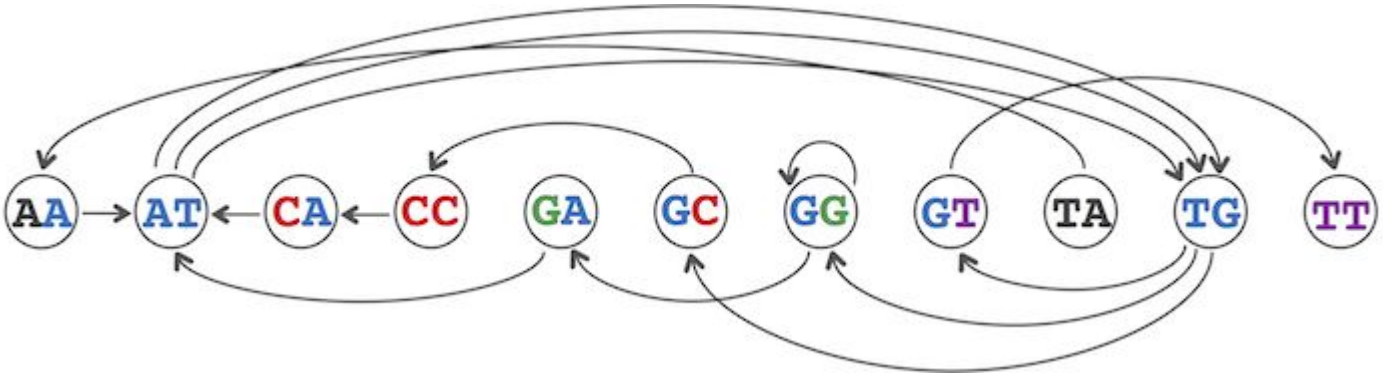
Algoritmo para construir grafo de De Bruijn

```
DeBruijn(Patterns)
    dB ← graph in which every k-mer in Patterns is isolated edge between its prefix and suffix
    dB ← graph resulting from gluing all nodes in dB with identical labels
    return dB
```

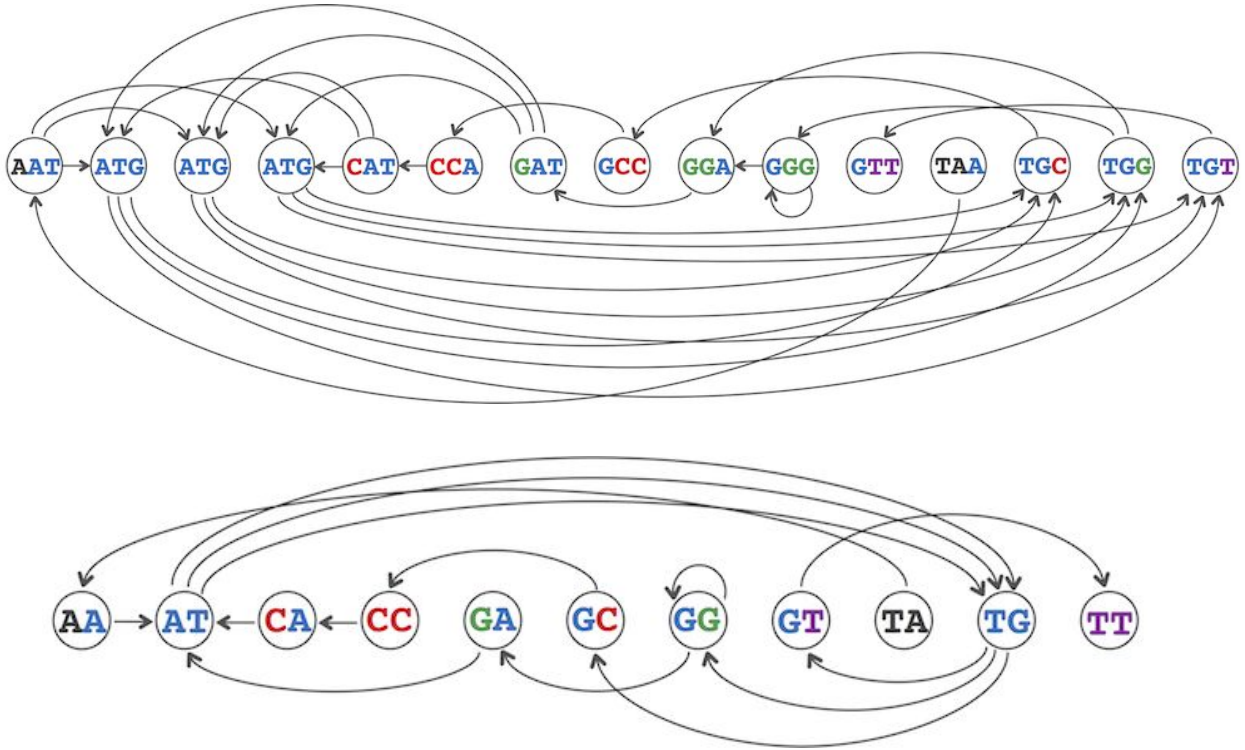
Otra forma de construir el grafo de De Bruijn

AAT ATG ATG ATG CAT CCA GAT GCC GGA GGG GTT TAA TGC
TGG TGT

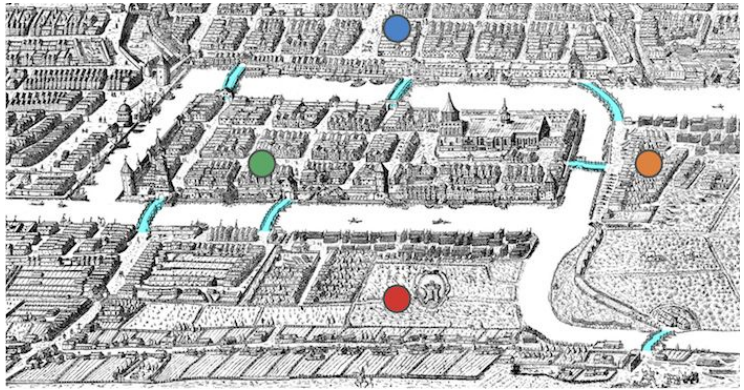
AA AT CA CC GA GC GG GT TA TG TT

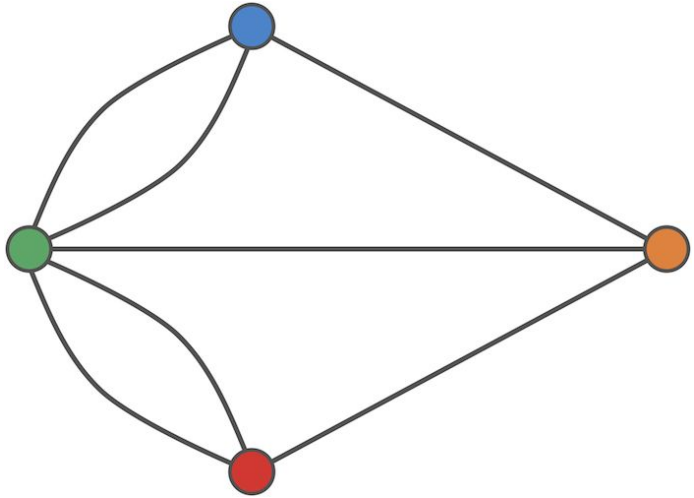


Camino Hamiltoniano vs Camino Euleriano

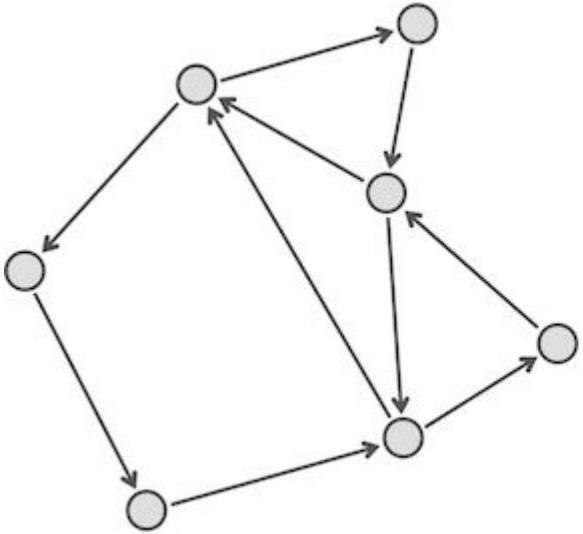


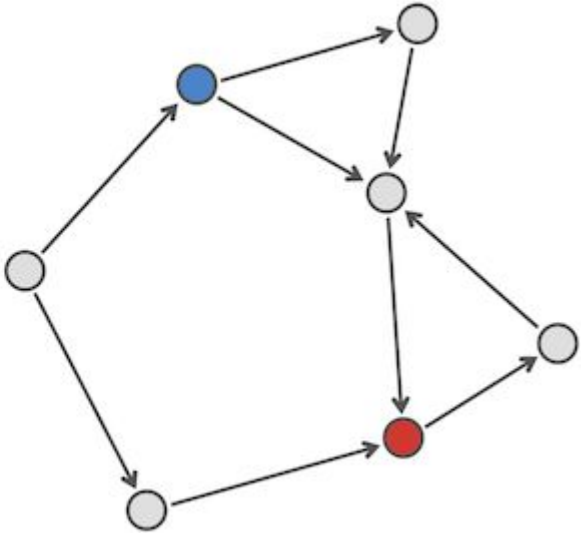
Ciclos eulerianos



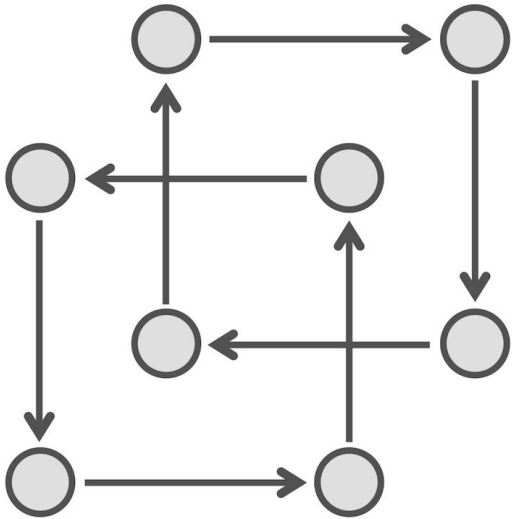


¿Por qué hay grafos que no tienen ciclos eulerianos?





Conexidad en grafos

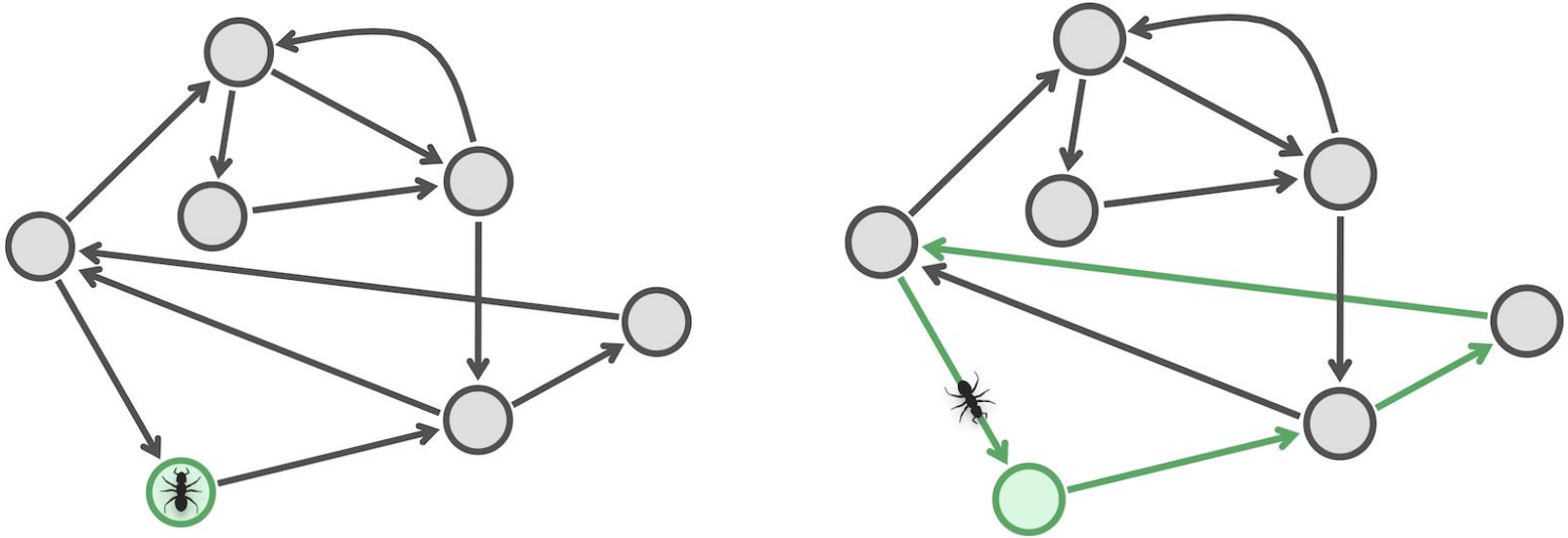


Un resultado

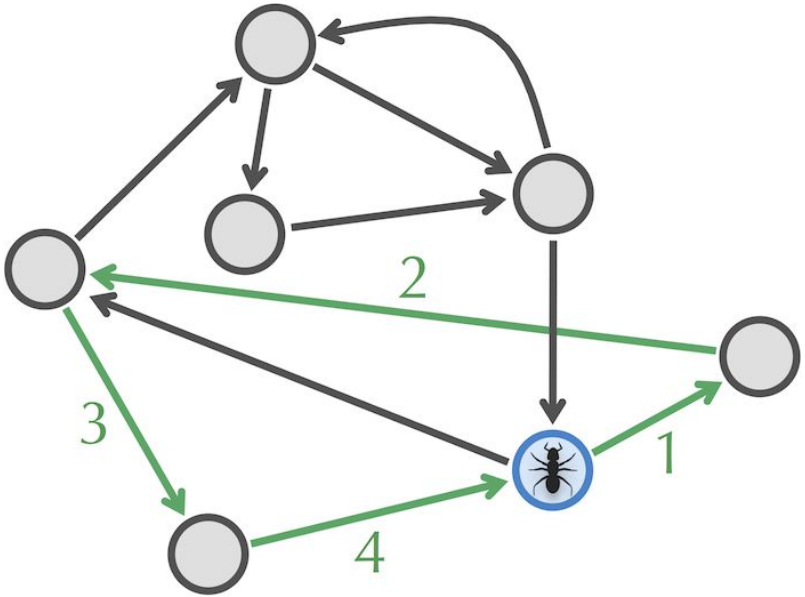
Teorema de Euler: Todo grafo dirigido balanceado fuertemente conexo es Euleriano, esto es, contiene un ciclo euleriano.

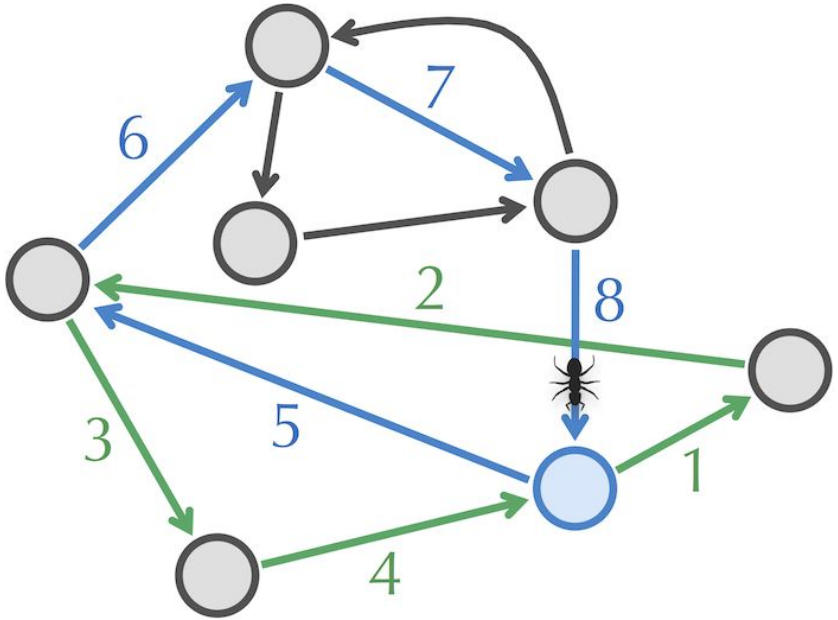
Veamos a continuación cómo construir o encontrar dicho ciclo.

Un ejemplo

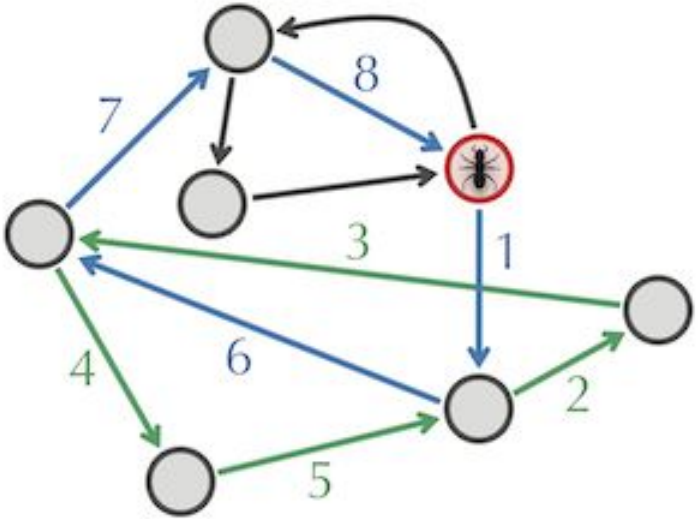


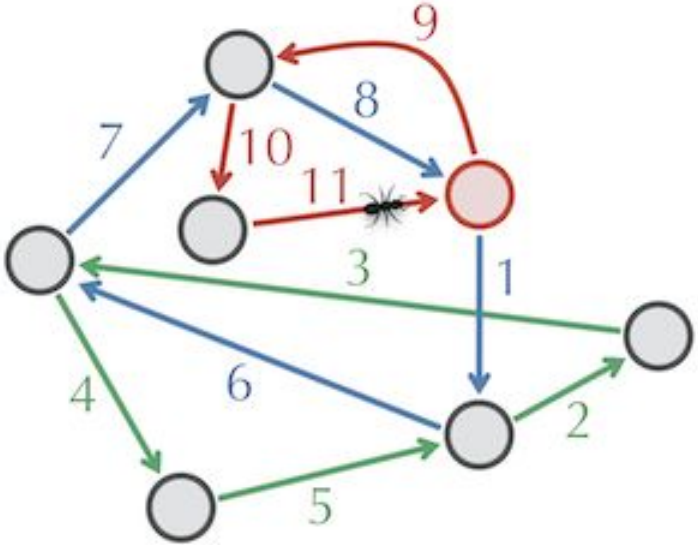
Un ejemplo





Un ejemplo

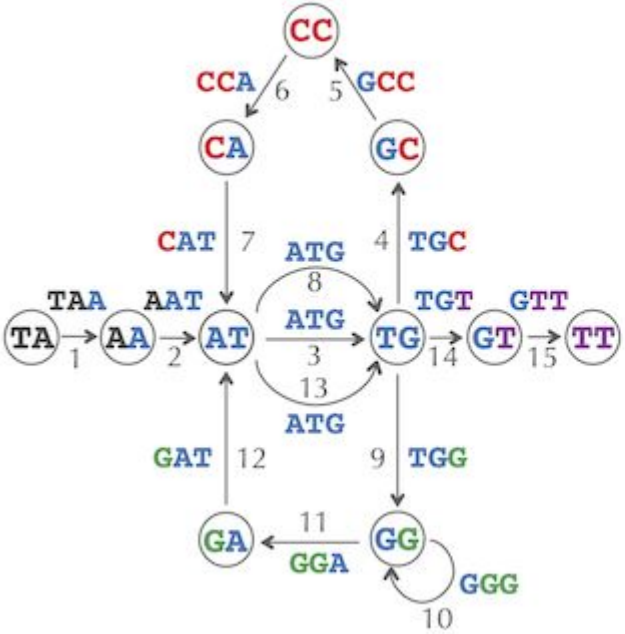


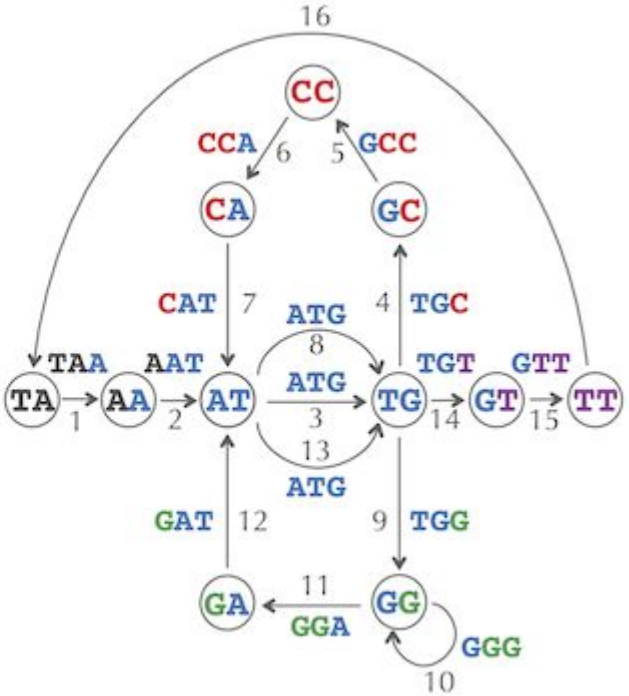


Algoritmo para construir ciclo euleriano

```
EulerianCycle(Graph)
    form a cycle Cycle by randomly walking in Graph (don't visit the same edge twice!)
    while there are unexplored edges in Graph
        select a node newStart in Cycle with still unexplored edges
        form Cycle' by traversing Cycle (starting at newStart) and then randomly walking
        Cycle ← Cycle'
    return Cycle
```

Nuestro caso: un grafo **casi balanceado**





Resumen:

```
StringReconstruction(Patterns)
    dB ← DeBruijn(Patterns)
    path ← EulerianPath(dB)
    Text ← PathToGenome(path)
    return Text
```

Algunas cosas a considerar