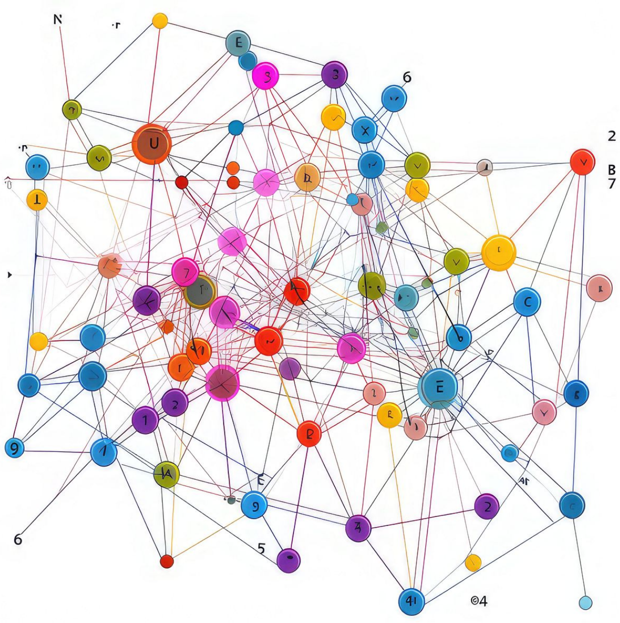
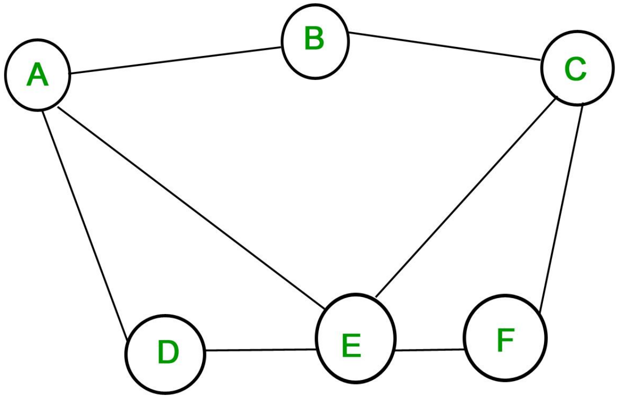
Montaje de secuencias

Grafos o gráficas





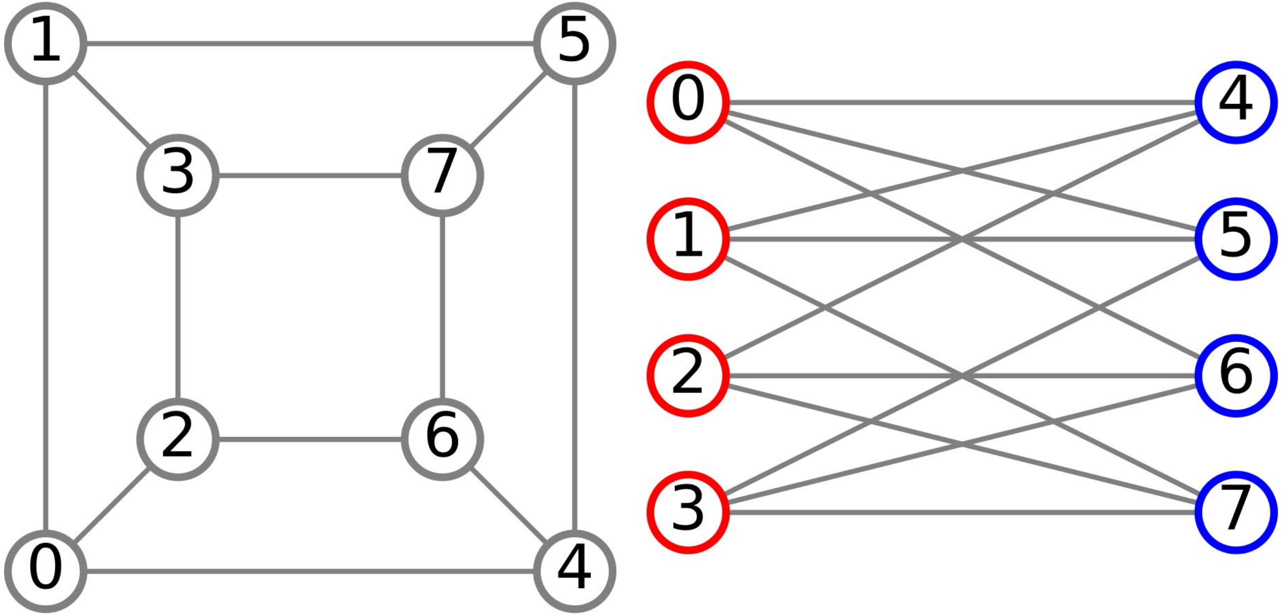
¿Qué es un grafo?

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).

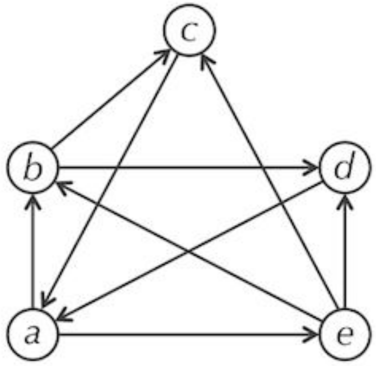
Puentes de Königsberg

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

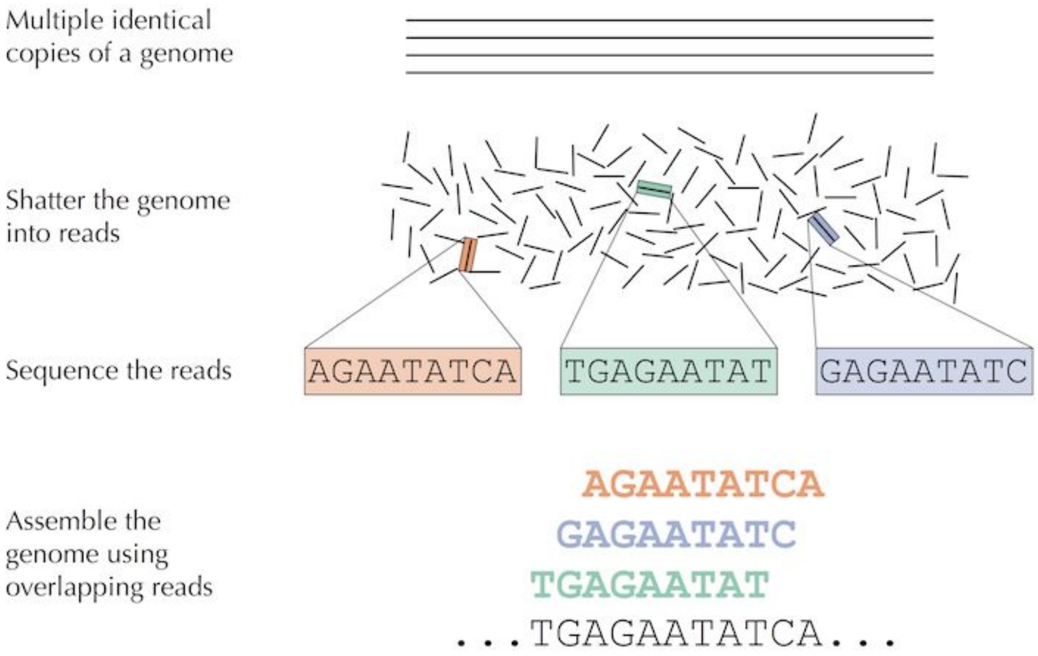
Adjacency List

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
AGAATATCA
TGAGAATAT
GAGAATATC
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.

$$\mathit{Composition}_3(\text{TATGGGTGC}) = \{\text{ATG, GGG, GGG, GGT, GTG, TAT, TGC, TGG}\}.$$

Reconstrucción de una cadena

AAT ATG GTT TAA TGT

TAA
AAT
ATG
TGT
GTT
TAATGTT

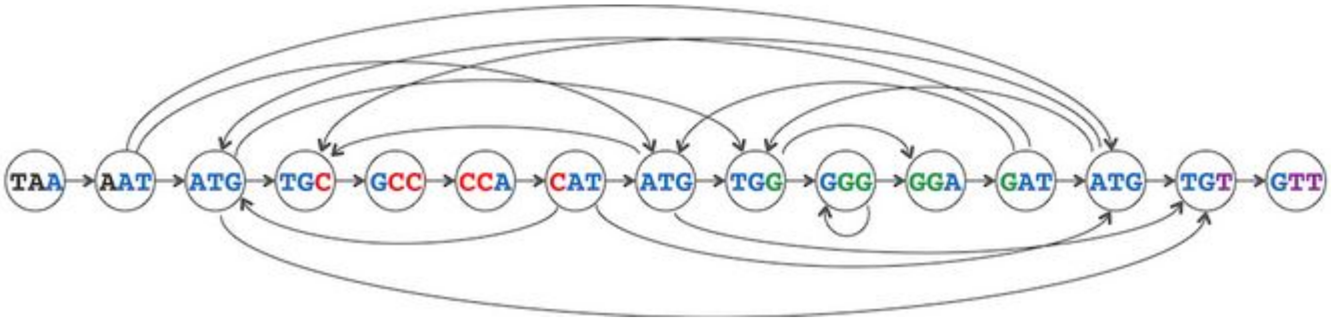
Una reconstrucción más larga

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

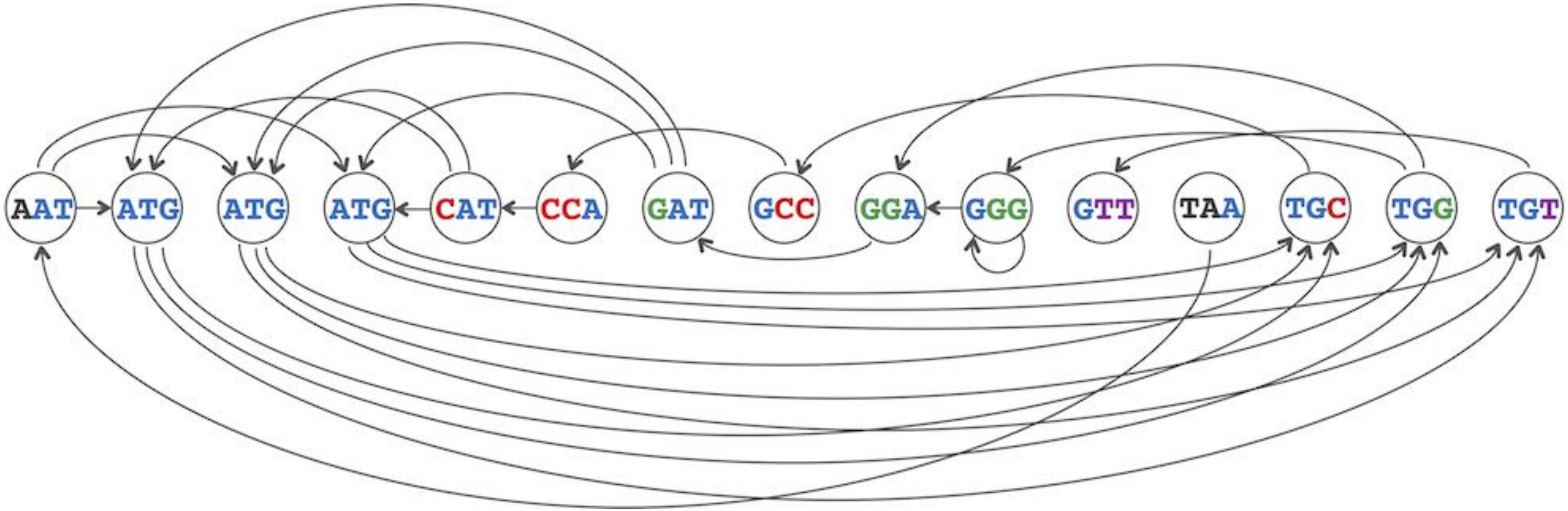
TAA
AAT
ATG
TGC
GCC
CCA
CAT
ATG
TGG
GGA
GAT
ATG
TGT
GTT

Utilizamos grafos

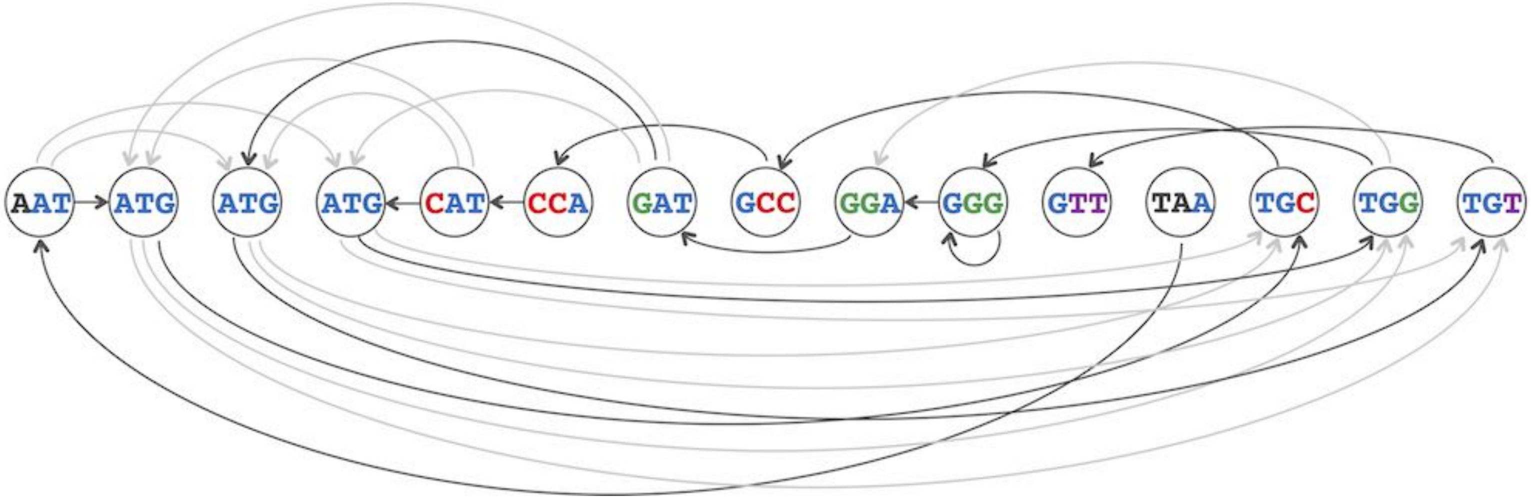
Grafo dirigido



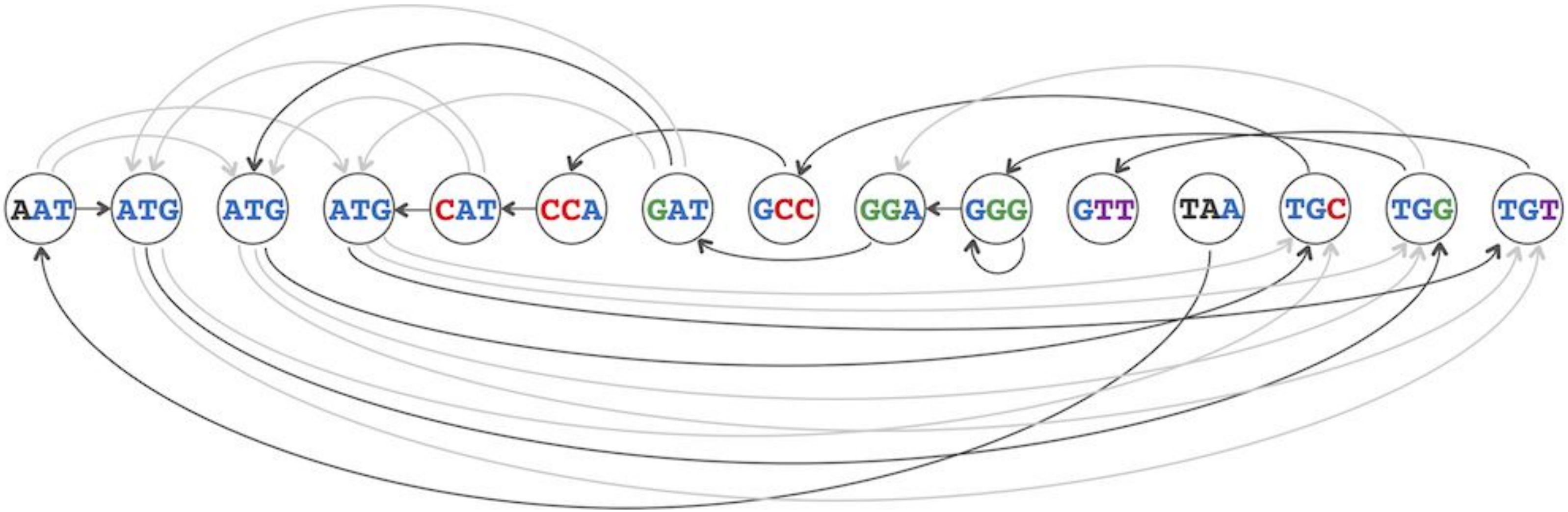
Si desordenamos un poco…



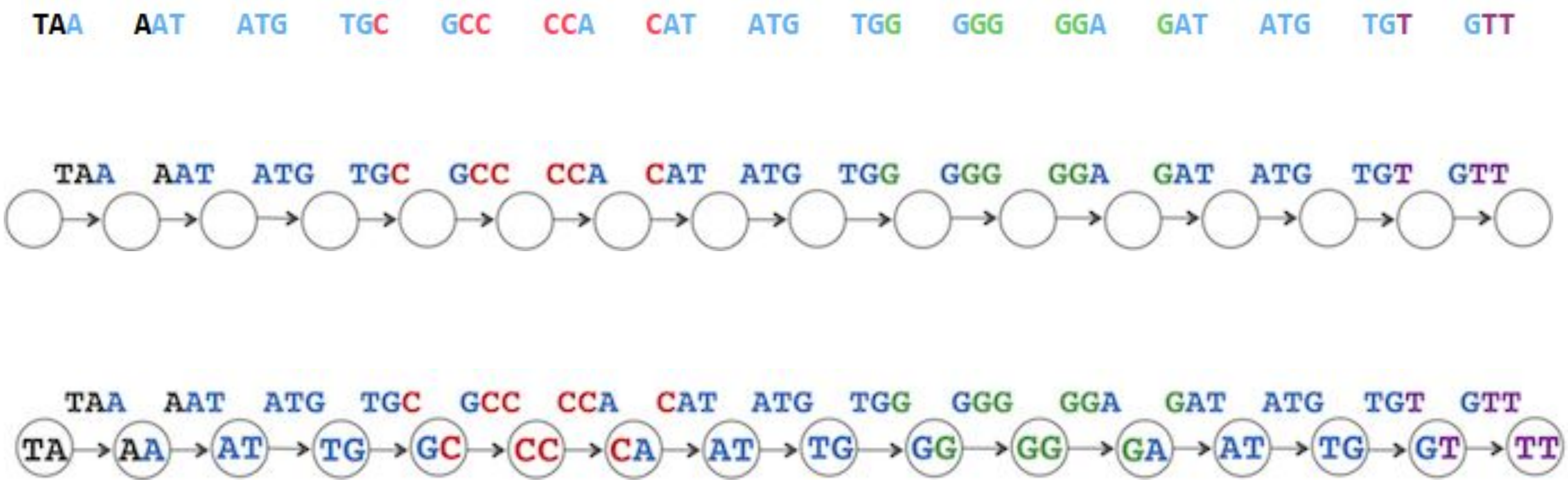
Si desordenamos un poco…



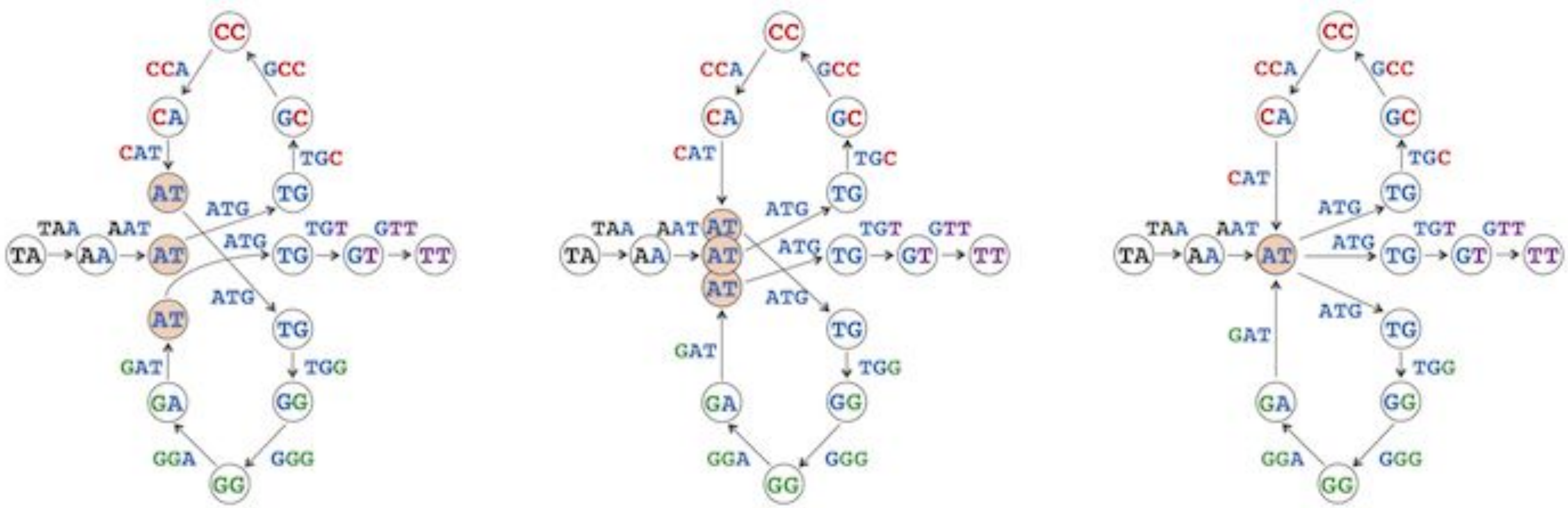
Caminos Hamiltonianos



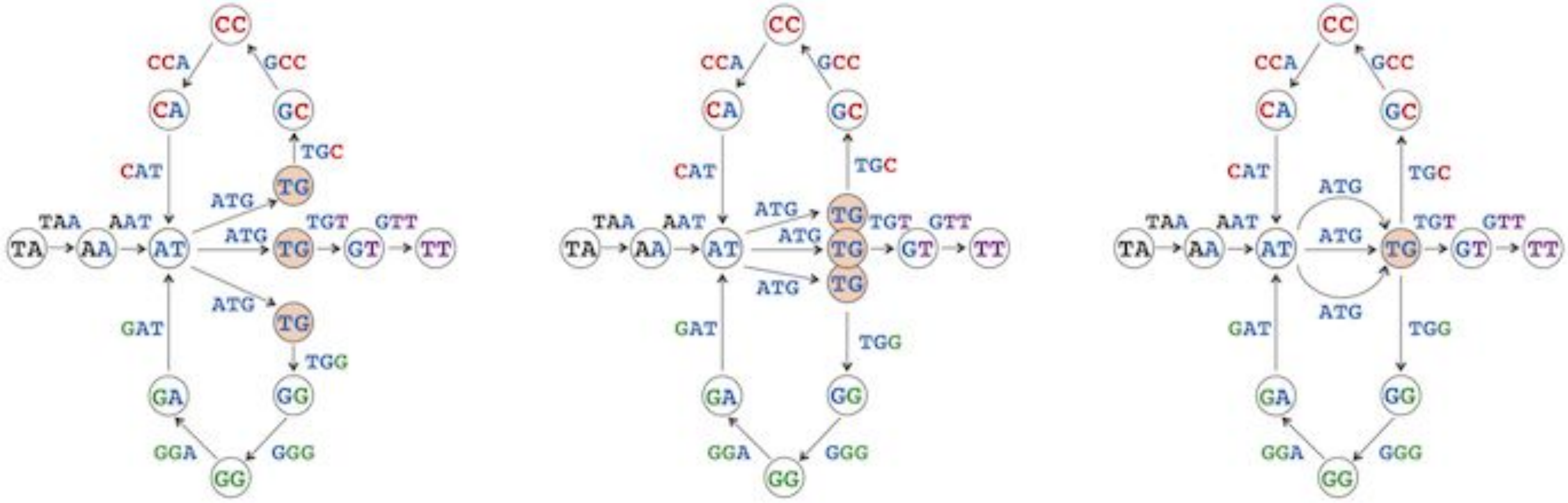
Otra forma de construir grafos



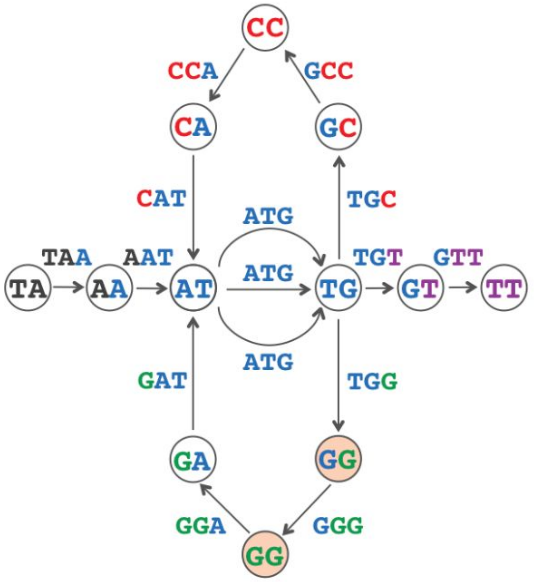
Transformamos el grafo

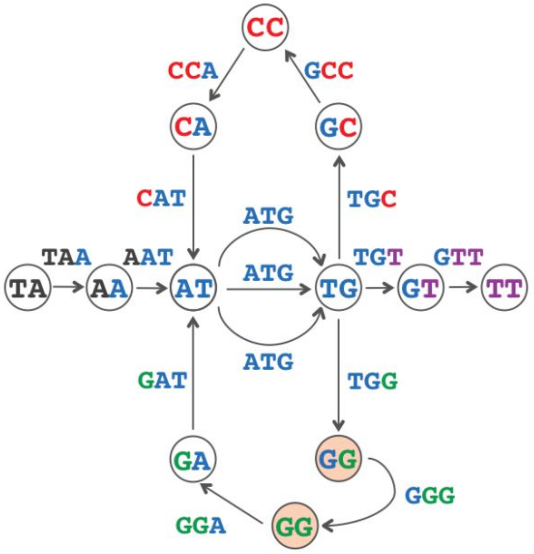


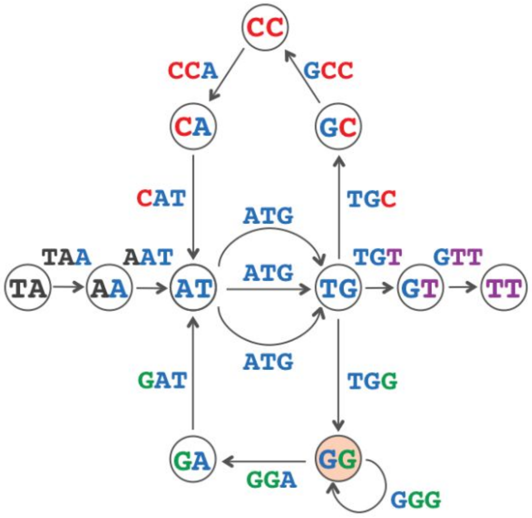
Transformamos el grafo



Transformamos el grafo

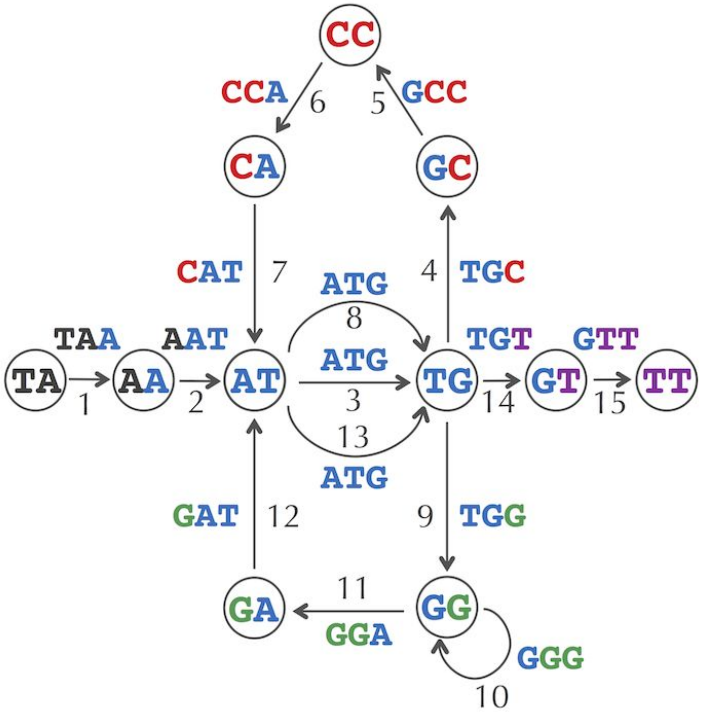




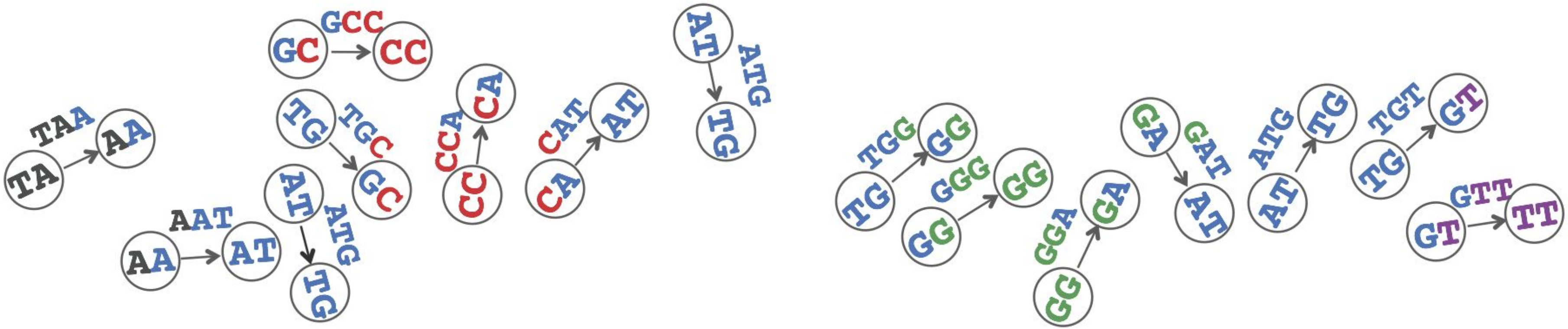


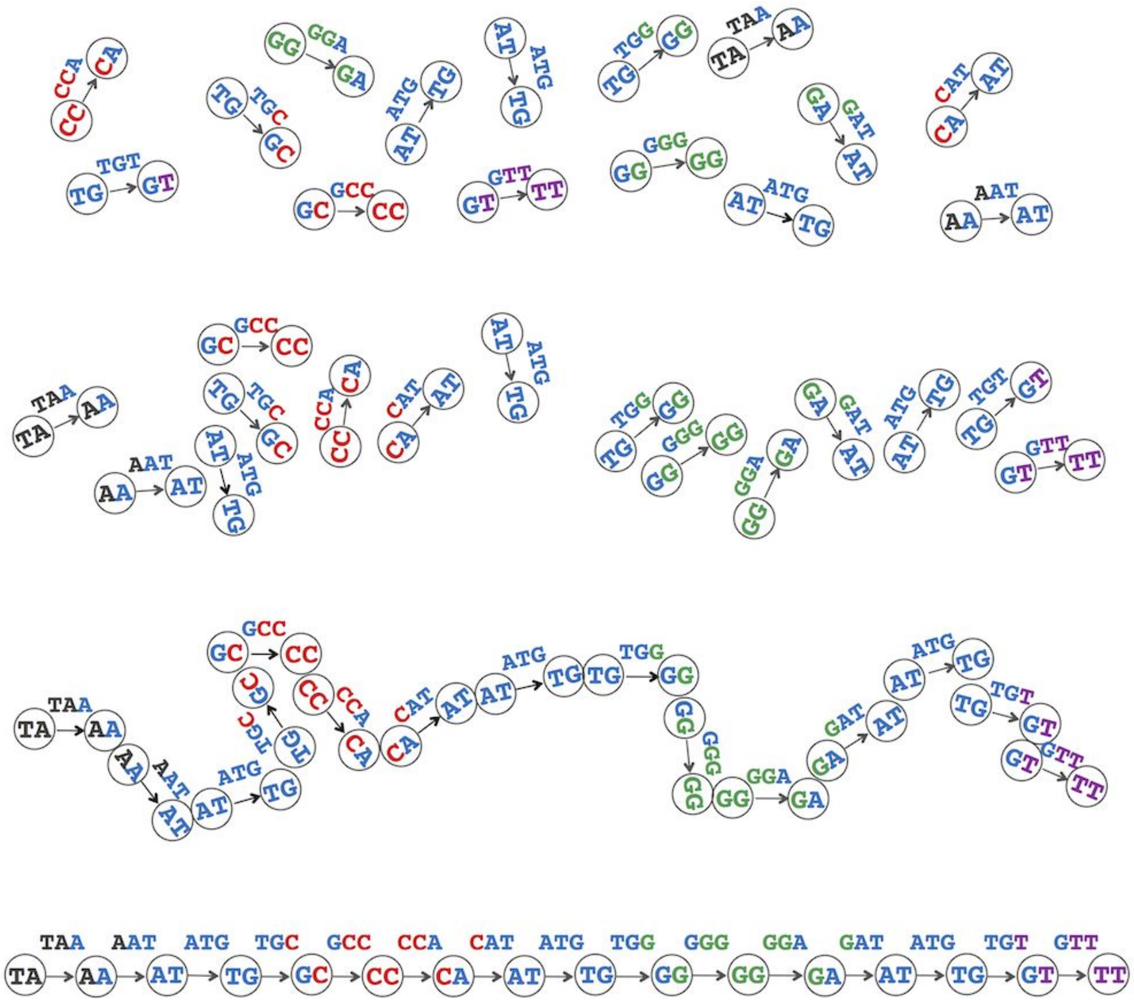
Caminos eulerianos

"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

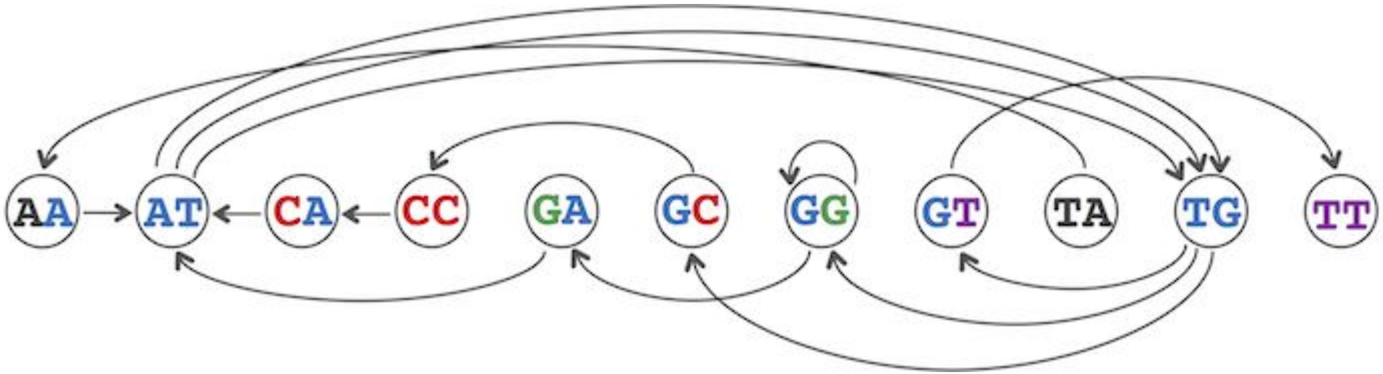
Algoritmo para construir grafo de De Bruijn

```
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
```

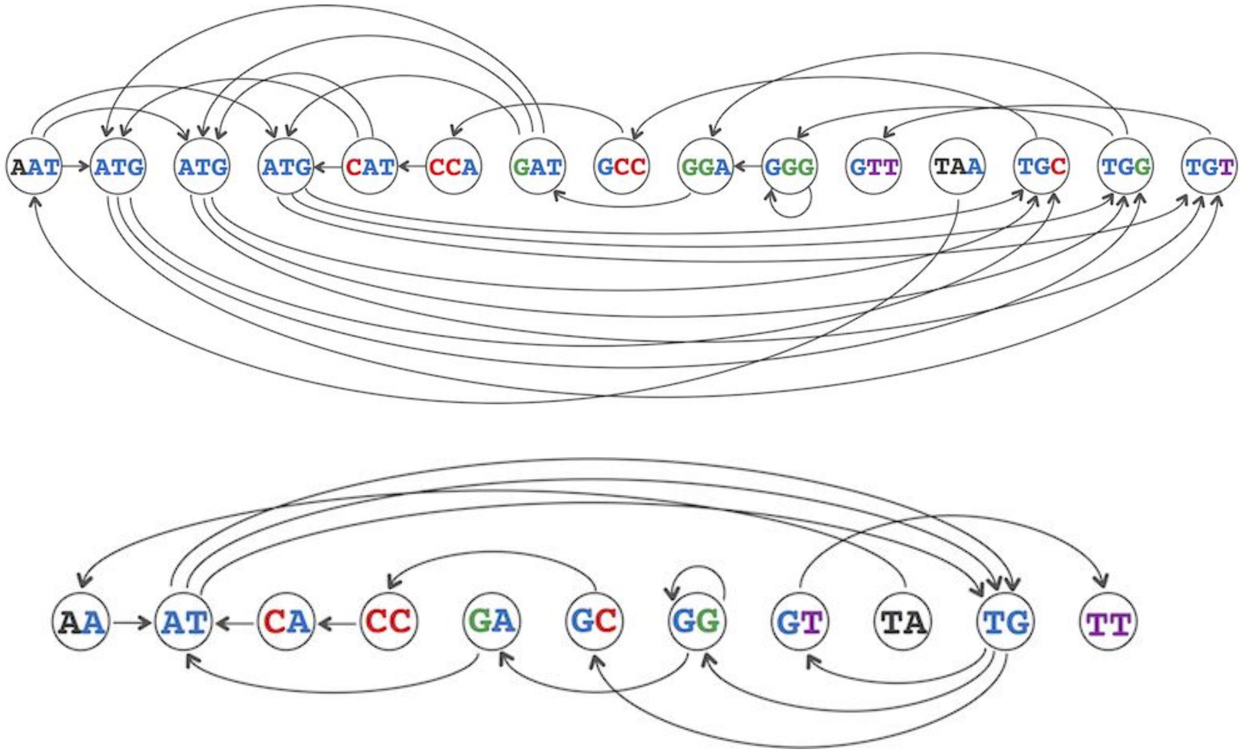
Otra forma de construir el grafo de De Bruijn

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

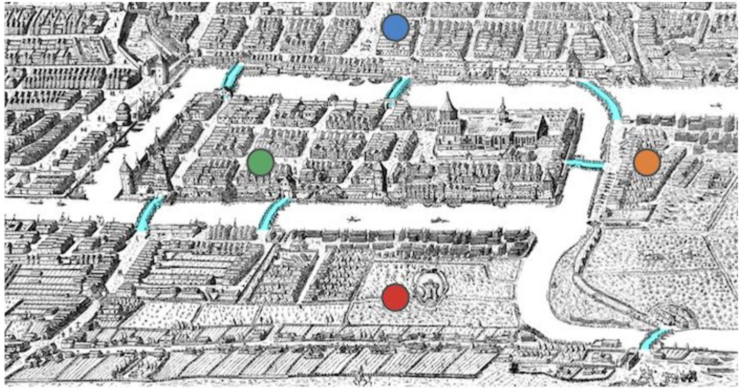
AA AT CA CC GA GC GG GT TA TG TT

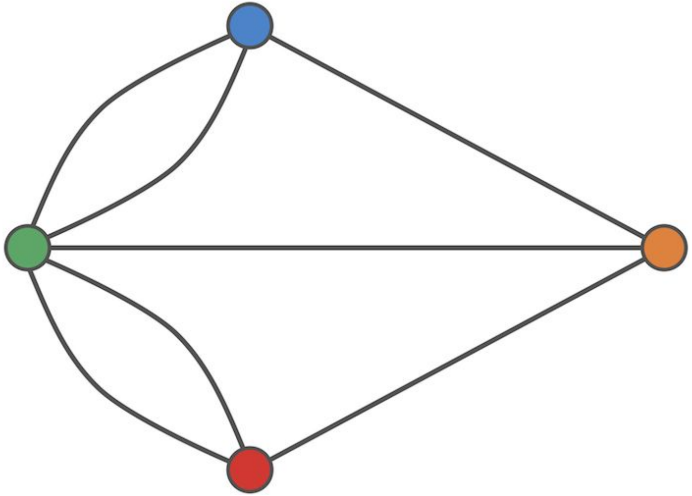


Camino Hamiltoniano vs Camino Euleriano

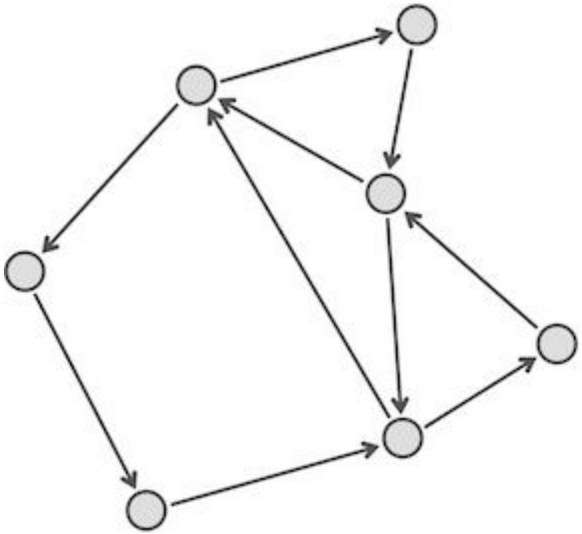


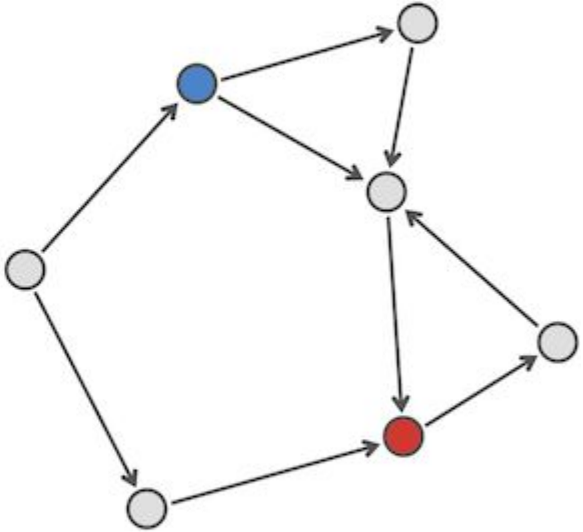
Ciclos eulerianos



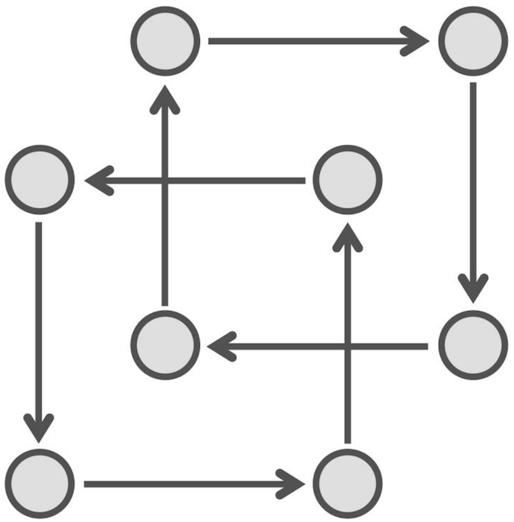


¿Por qué hay grafos que no tienen ciclos eulerianos?

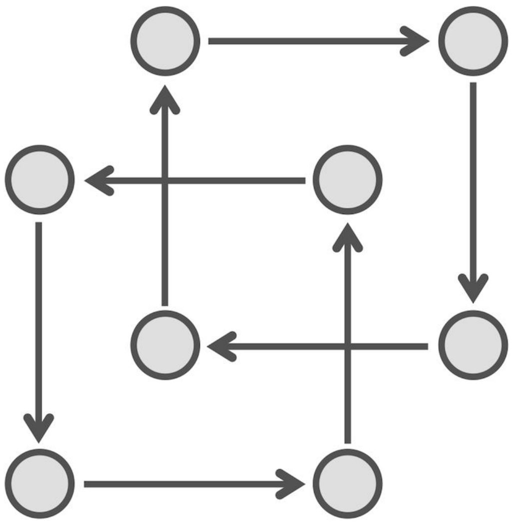




Conexidad en grafos



Conexidad en grafos

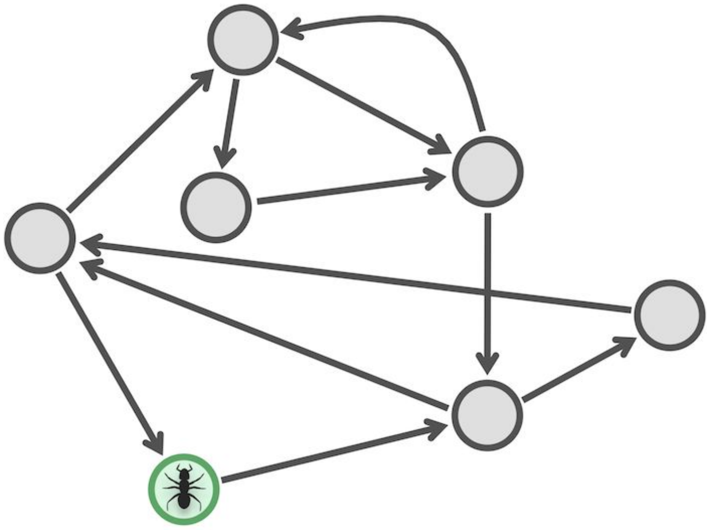


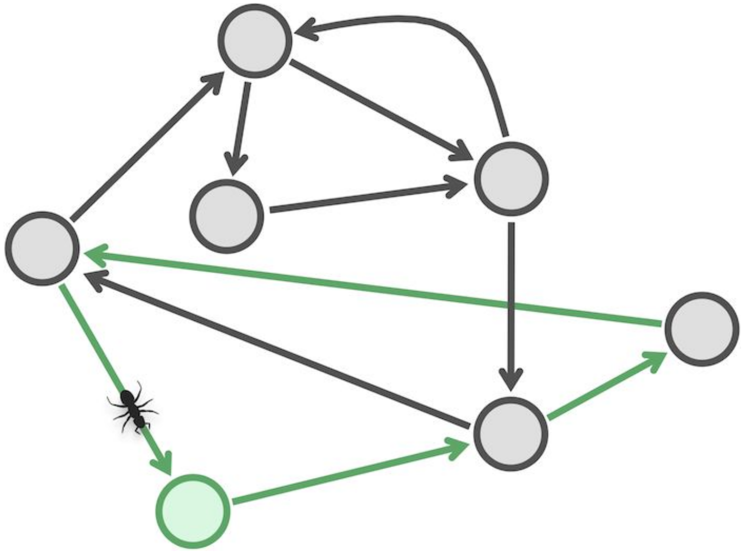
Un resultado

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

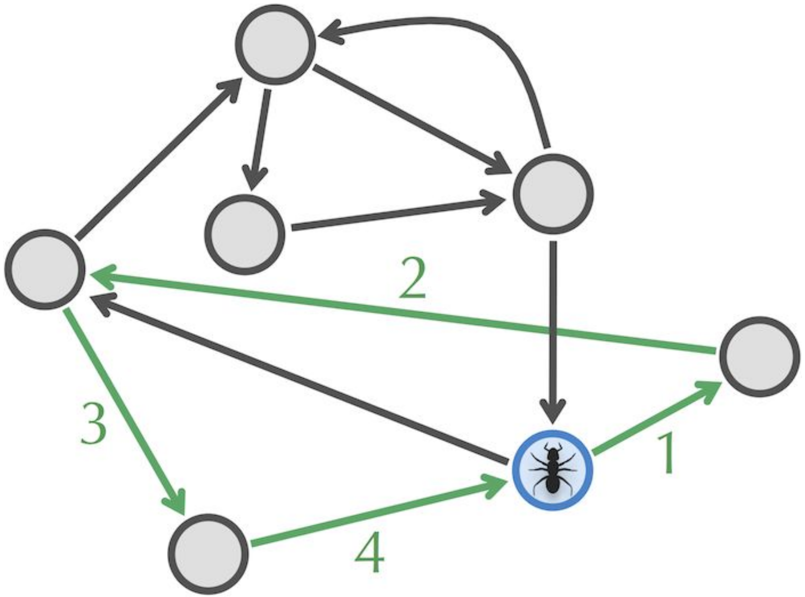
Veamos a continuación cómo construir o encontrar dicho ciclo.

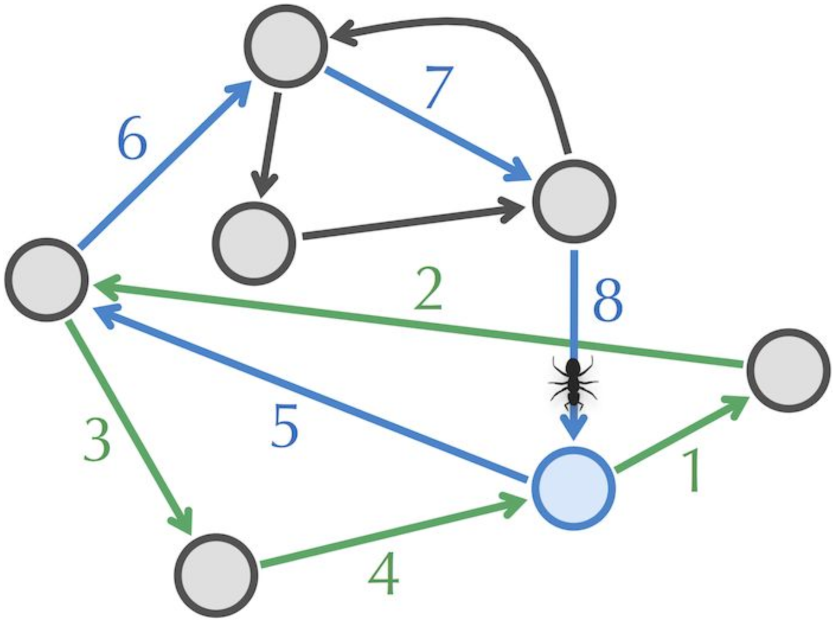
Un ejemplo



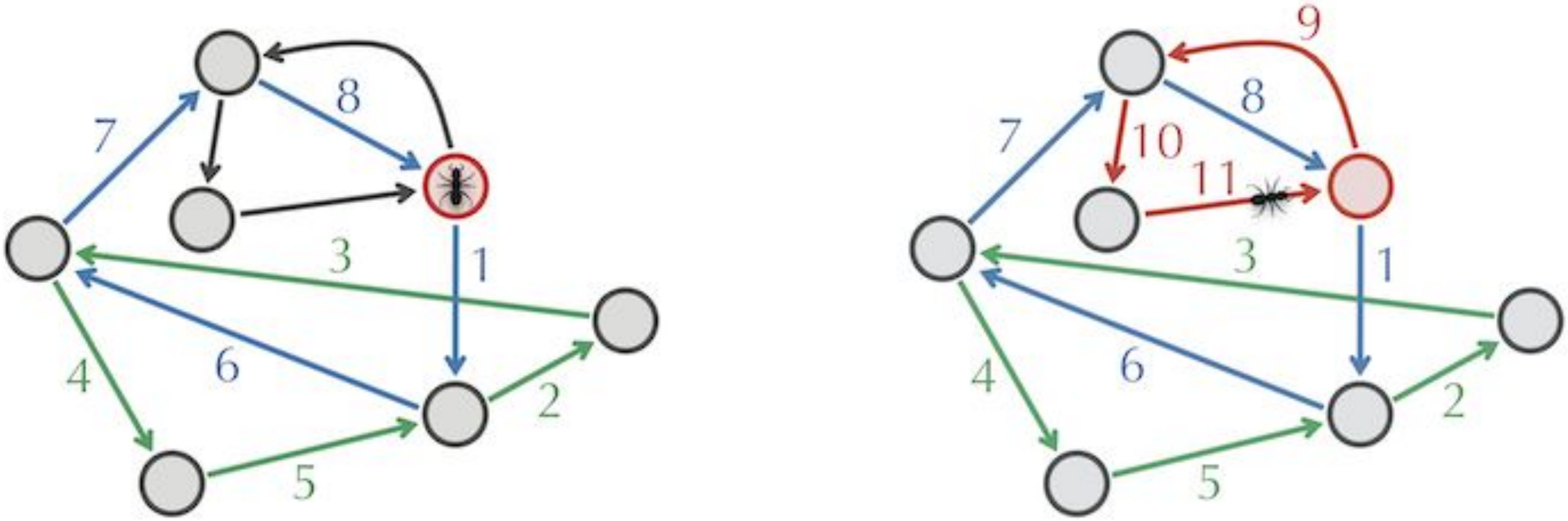


Un ejemplo





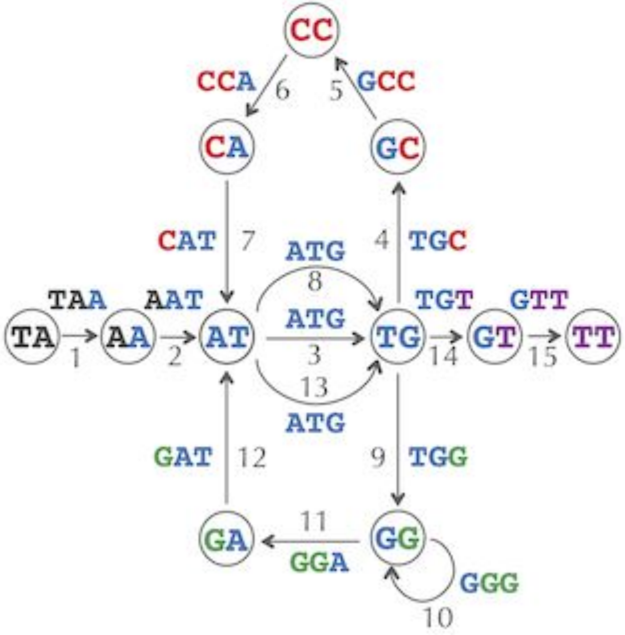
Un ejemplo

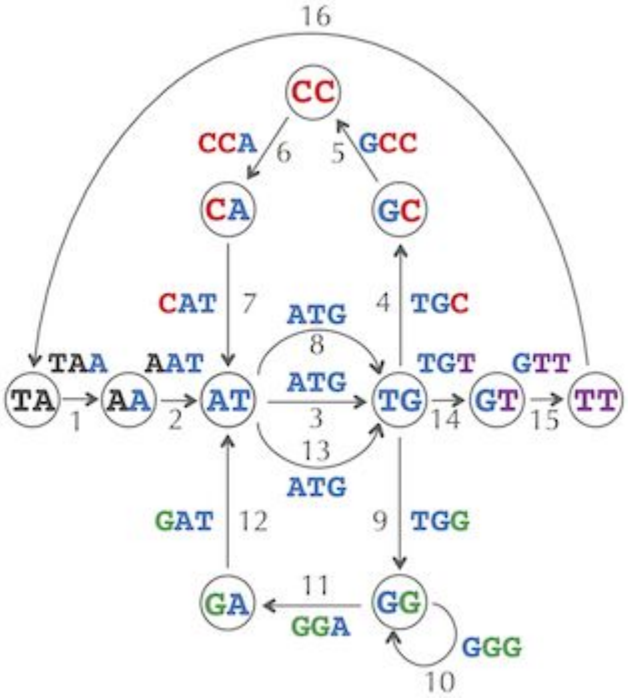


Algoritmo para construir ciclo euleriano

```
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**

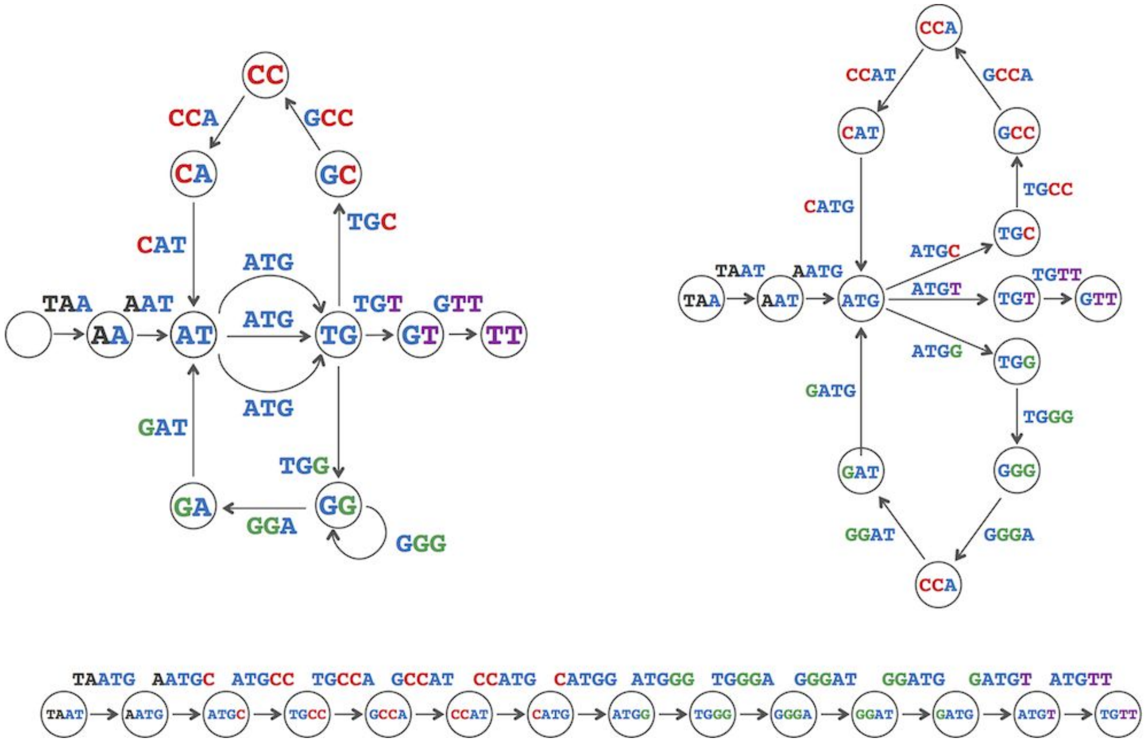




Resumen:

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

Algunas cosas a considerar



Actividad

Selecciona una secuencia trabajada.

Obtén la **composición** de dicha secuencia.

Investiga cómo encontrar el camino euleriano en un grafo utilizando la librería **networkx**. Con esta funcionalidad, implementa una función que reconstruye una cadena a partir del camino euleriano.

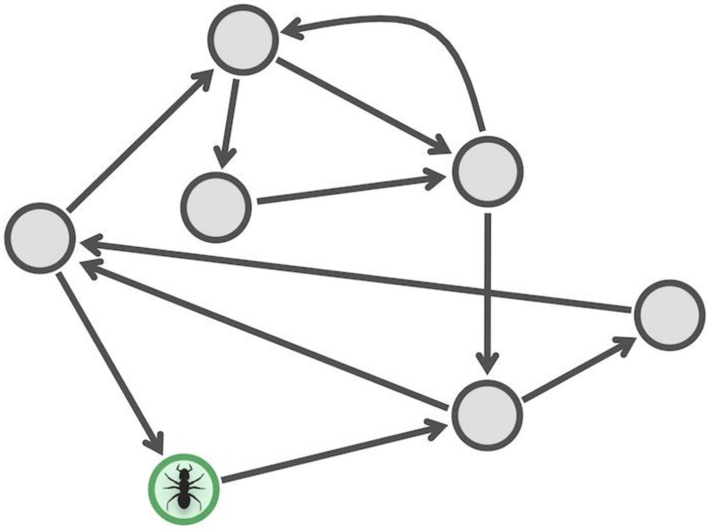
¿Obtuviste la cadena original?

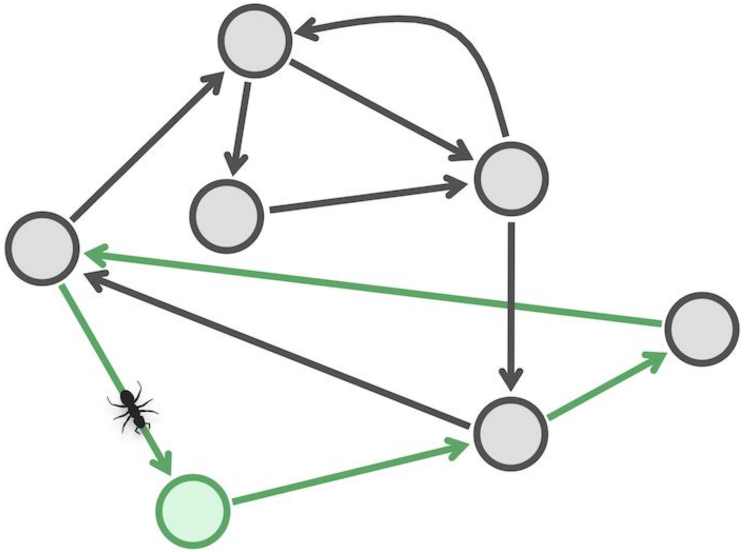
Un resultado

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

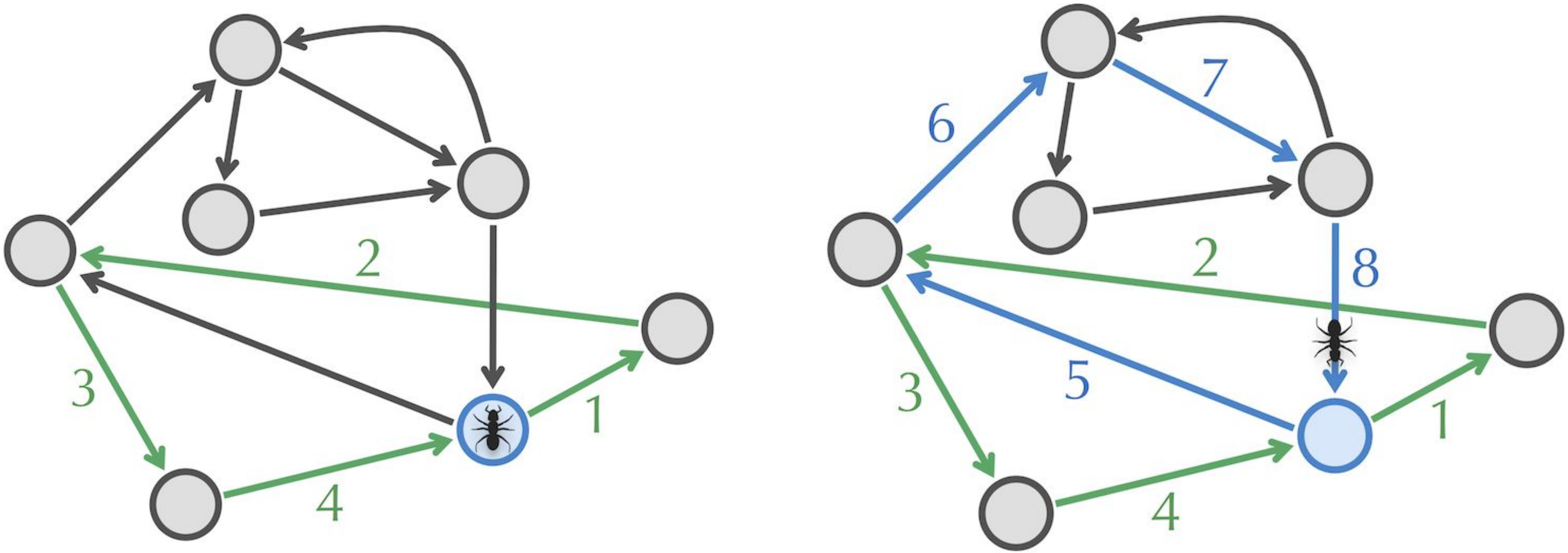
Veamos a continuación cómo construir o encontrar dicho ciclo.

Un ejemplo

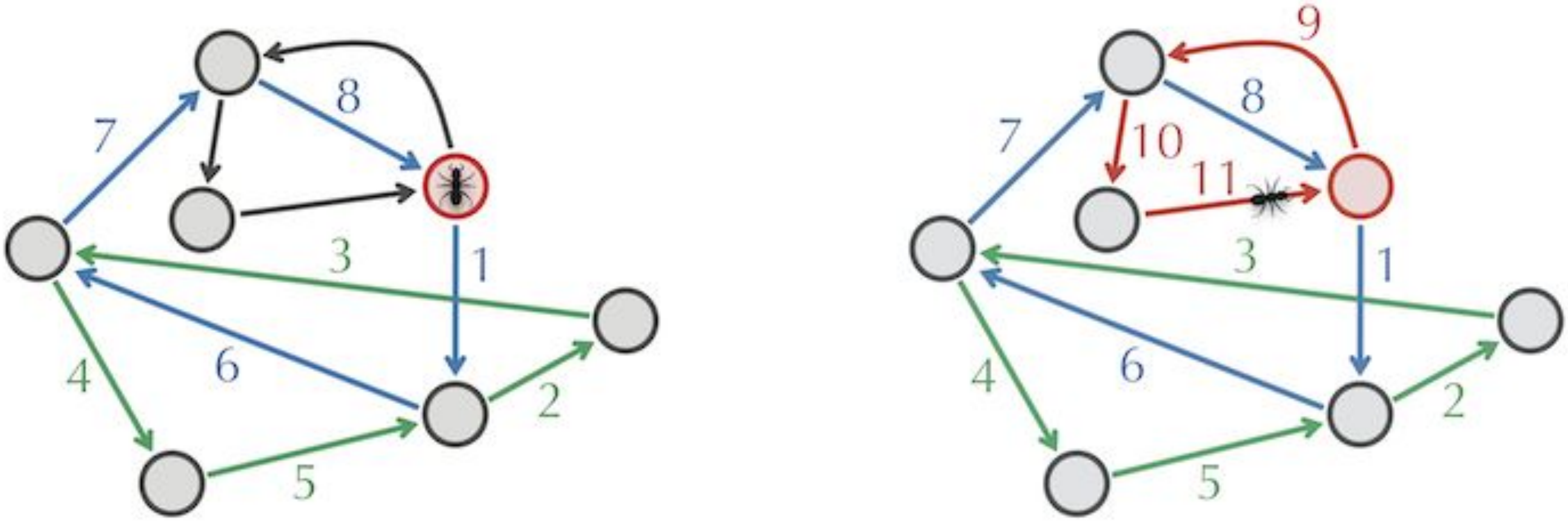




Un ejemplo



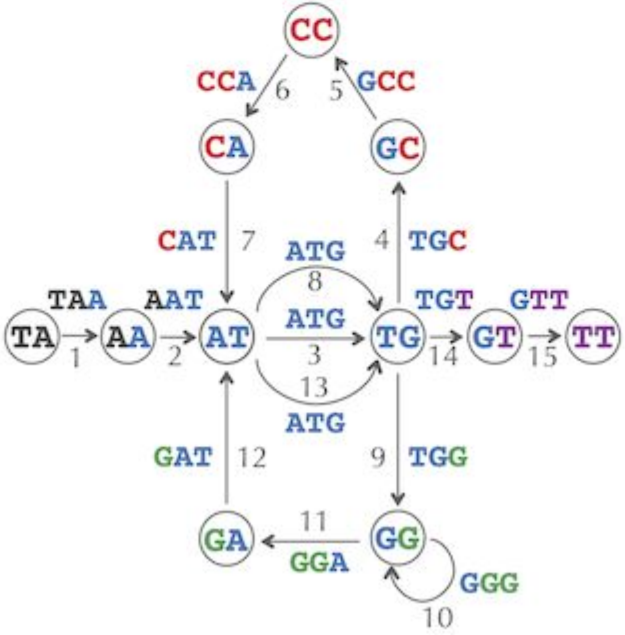
Un ejemplo

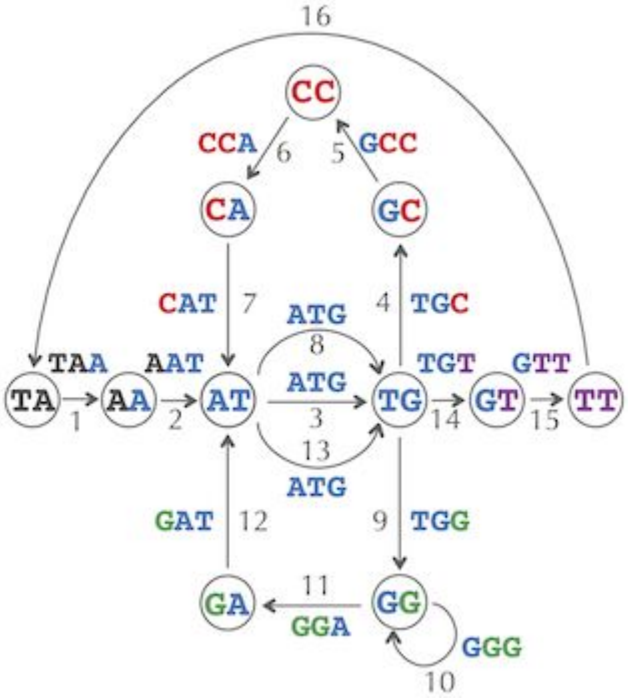


Algoritmo para construir ciclo euleriano

```
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**

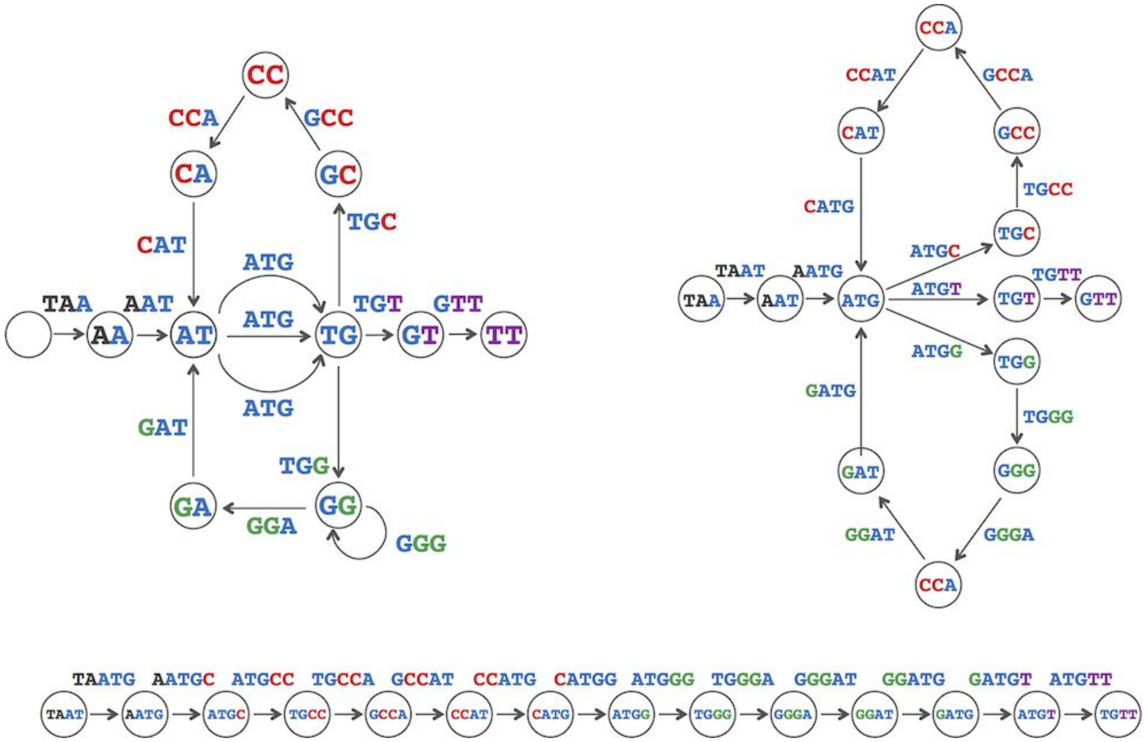




Resumen:

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

Algunas cosas a considerar



Actividad

Selecciona una secuencia trabajada.

Obtén la **composición** de dicha secuencia.

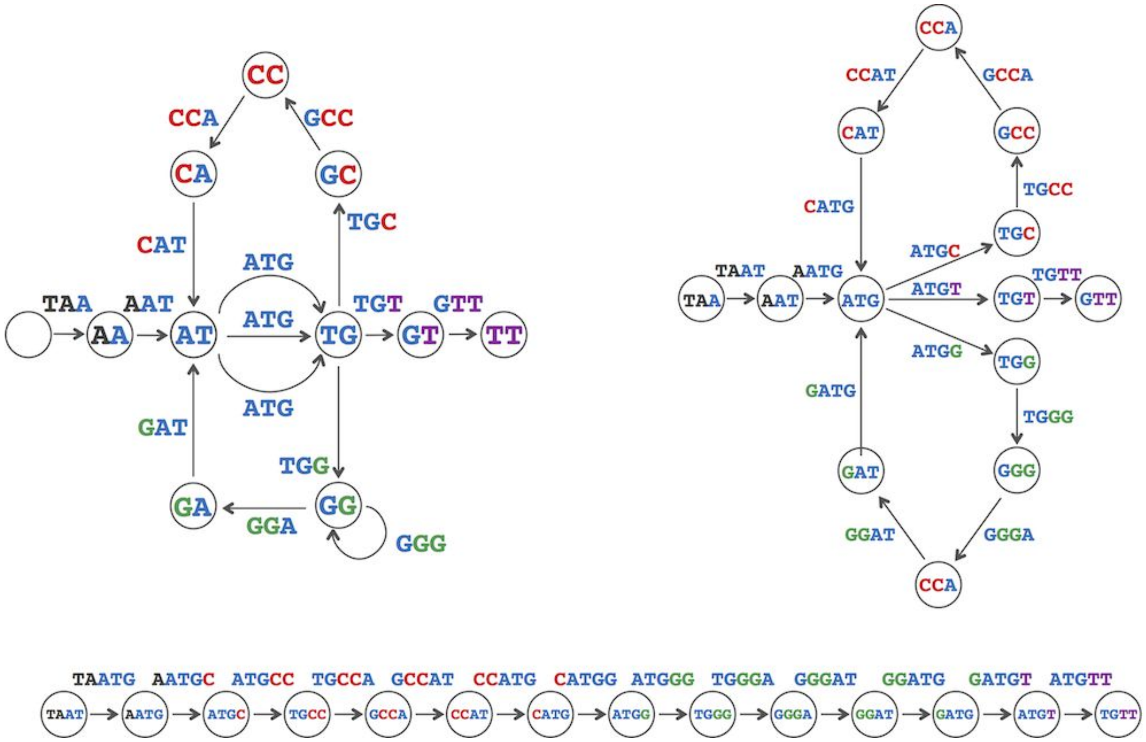
Investiga cómo encontrar el camino euleriano en un grafo utilizando la librería **networkx**. Con esta funcionalidad, implementa una función que reconstruye una cadena a partir del camino euleriano.

¿Obtuviste la cadena original?

Resumen:

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

Algunas cosas a considerar



Actividad

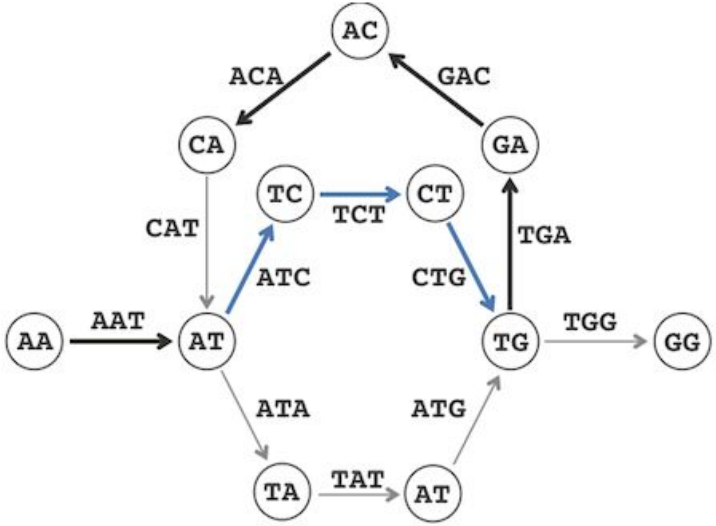
Selecciona una secuencia trabajada.

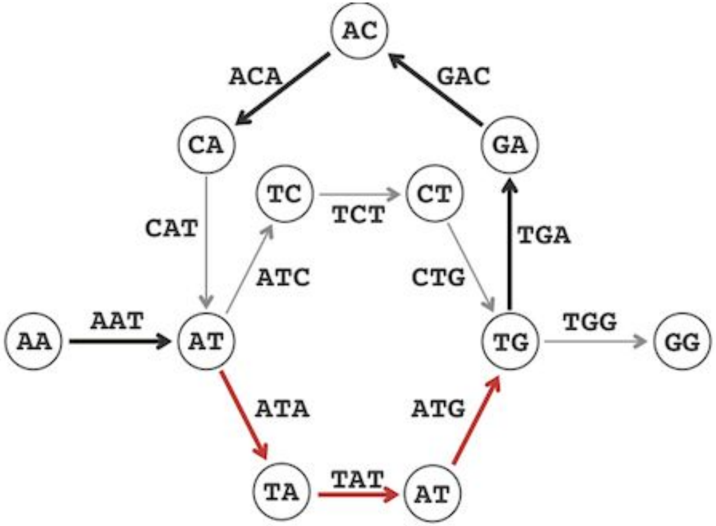
Obtén la **composición** de dicha secuencia.

Investiga cómo encontrar el camino euleriano en un grafo utilizando la librería **networkx**. Con esta funcionalidad, implementa una función que reconstruye una cadena a partir del camino euleriano.

¿Obtuviste la cadena original?

Relación entre un grafo de de Bruijn y una secuencia





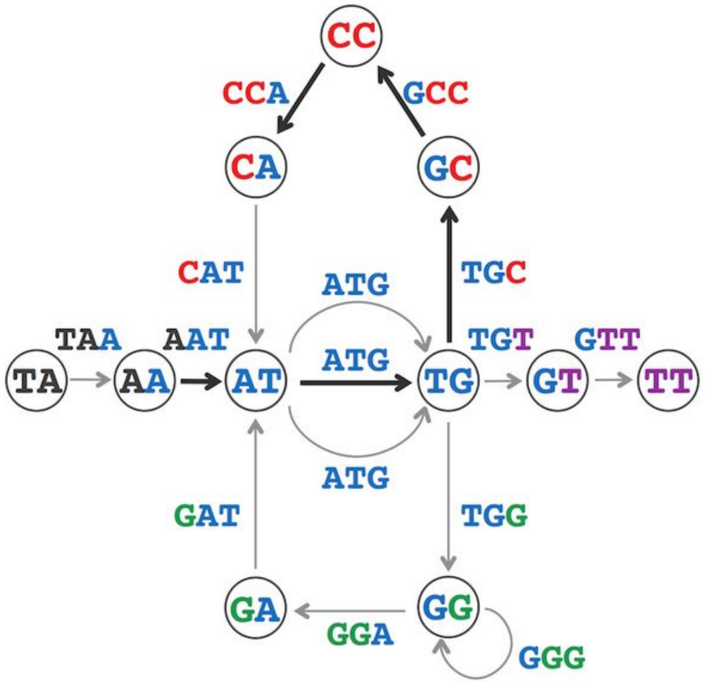
Lecturas (reads) por pares

Un (k,d)-mero es un par de k-meros separados por una distancia d. Utilizamos la notación $(Pattern_1|Pattern_2)$

Por ejemplo, (**AAT**|**TGG**) es un (3,4)-mero de **TAATGCCATGGGATGTT.**

La colección de todos los (k,d)-meros en una cadena se denota por $PairedComposition_{k,d}(Text)$,

Algunas ventajas de utilizar reads por pares



Composición de (k,d)-meros

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

Partes de un (k,d)-mero

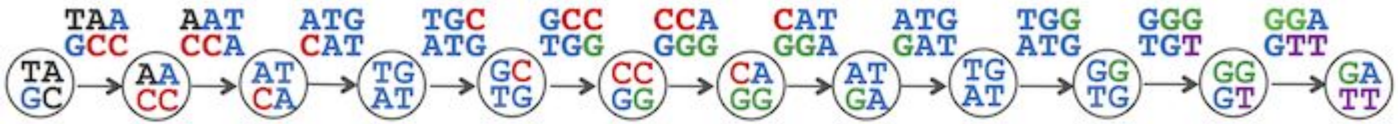
Dado un (k,d)-mero (a_1 ... a_k | b_1 ... b_k) definimos el prefijo como el (k-1,d+1)-mero:

$$(a_1 \ldots a_{k-1} \mid b_1 \ldots b_{k-1})$$

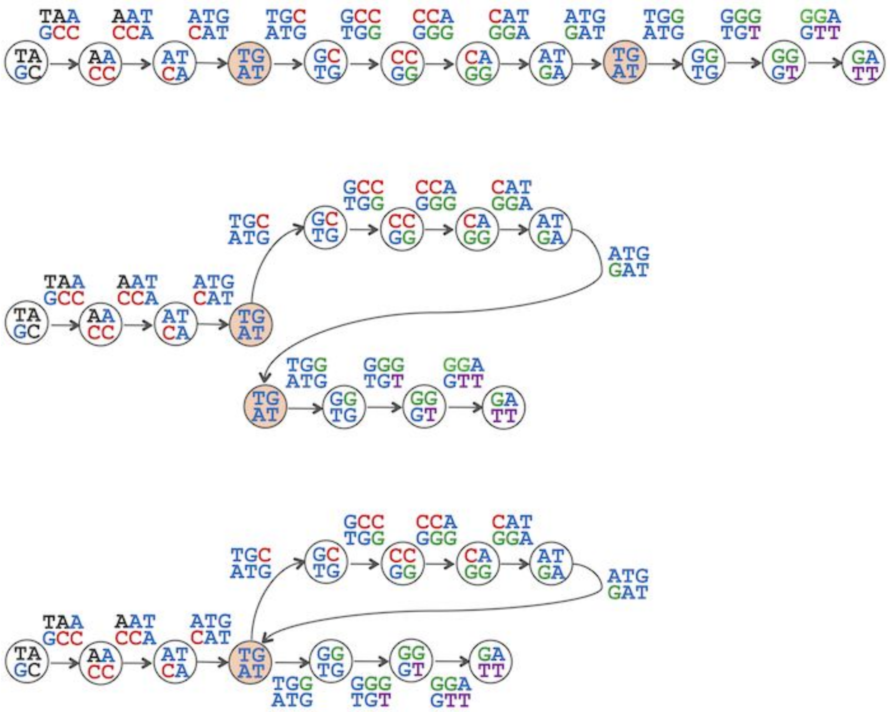
y el sufijo como el (k-1,d+1)-mero

$$(a_2 \ldots a_k \mid b_2 \ldots b_k)$$

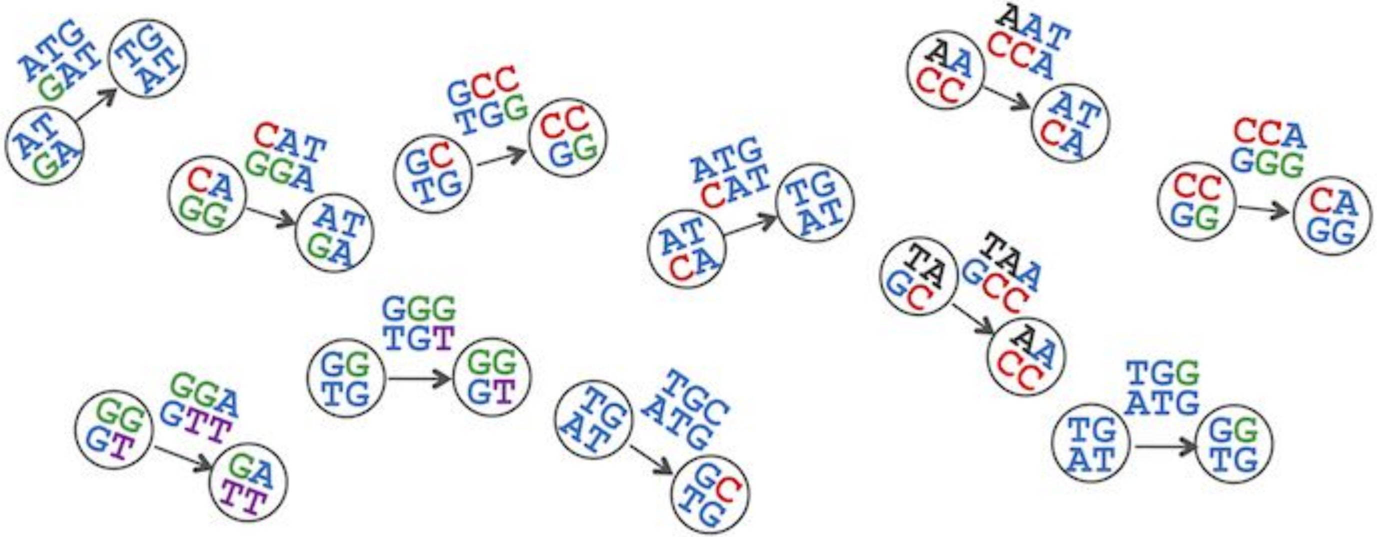
Grafo de de Bruijn con pares



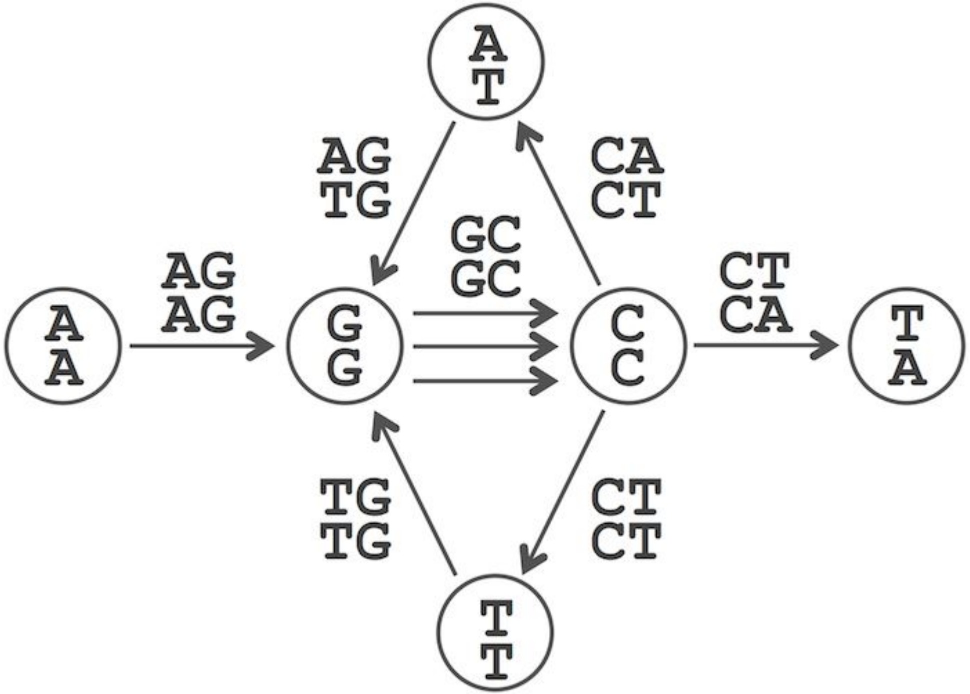
Construcción de grafo de de Bruijn con pares



Nuestros datos para el caso de lecturas en pares



Algunos problemas con las lecturas en par



Ensamblaje utilizando pares

```
AG-AG
 GC-GC
  CA-CT
   AG-TG
    GC-GC
     CT-CT
      TG-TG
       GC-GC
        CT-CA
AGCAGCTGCTGCA
```

Posibles errores…

```
AG-AG
 GC-GC
  CT-CT
   TG-TG
    GC-GC
     CA-CT
      AG-TG
       GC-GC
        CT-CA
AGC?GC?GCTGCA
```

Última actividad

1. Genera una función que, dada una cadena, construya la composición k,d.

2. Genera una función que dada la composición de (k,d)-meros, construya las aristas del grafo de De Bruijn.

3. Construye un grafo de De Bruijn y encuentra el ciclo euleriano.