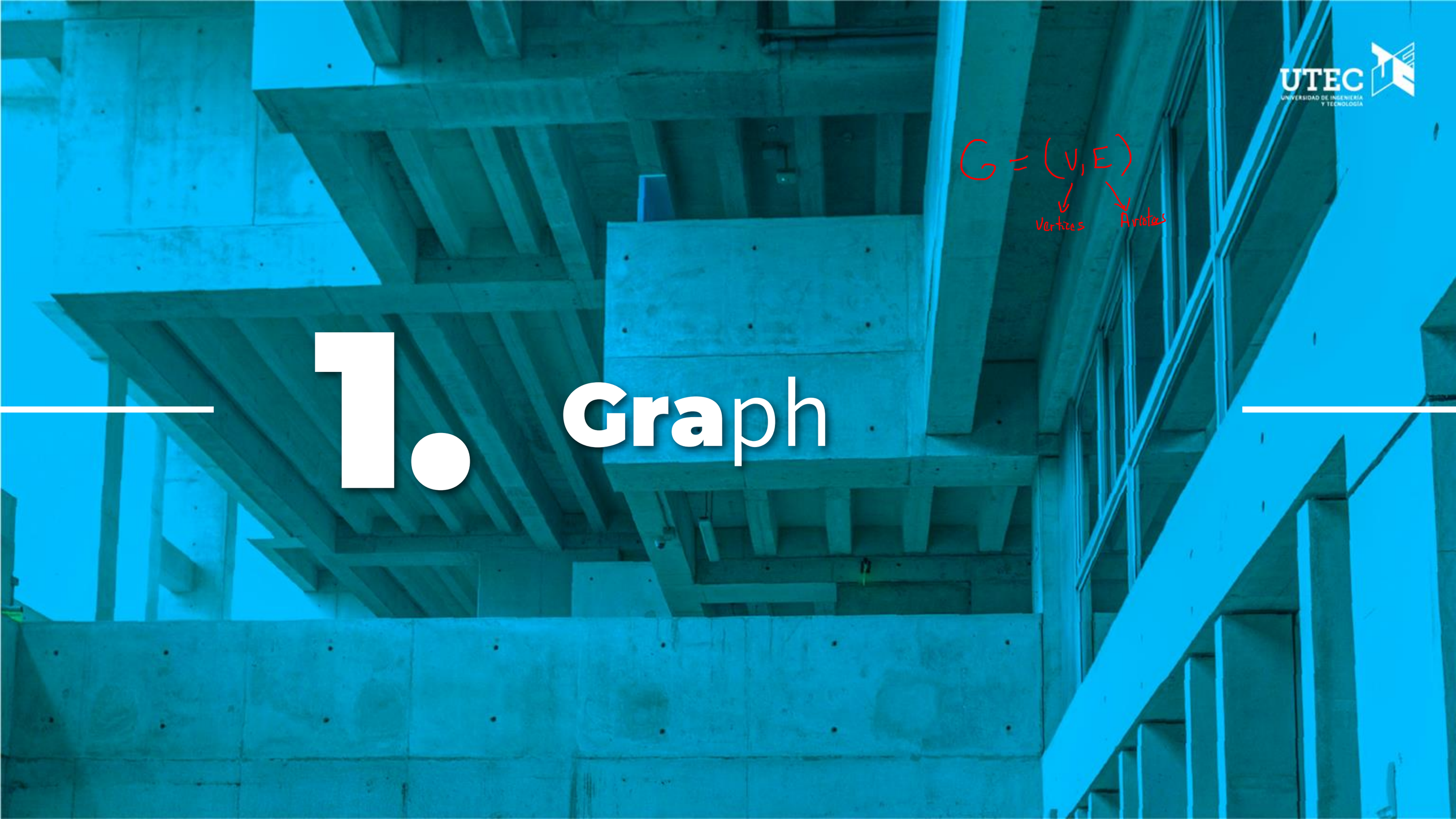
UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

OVA 3.0: Graphs

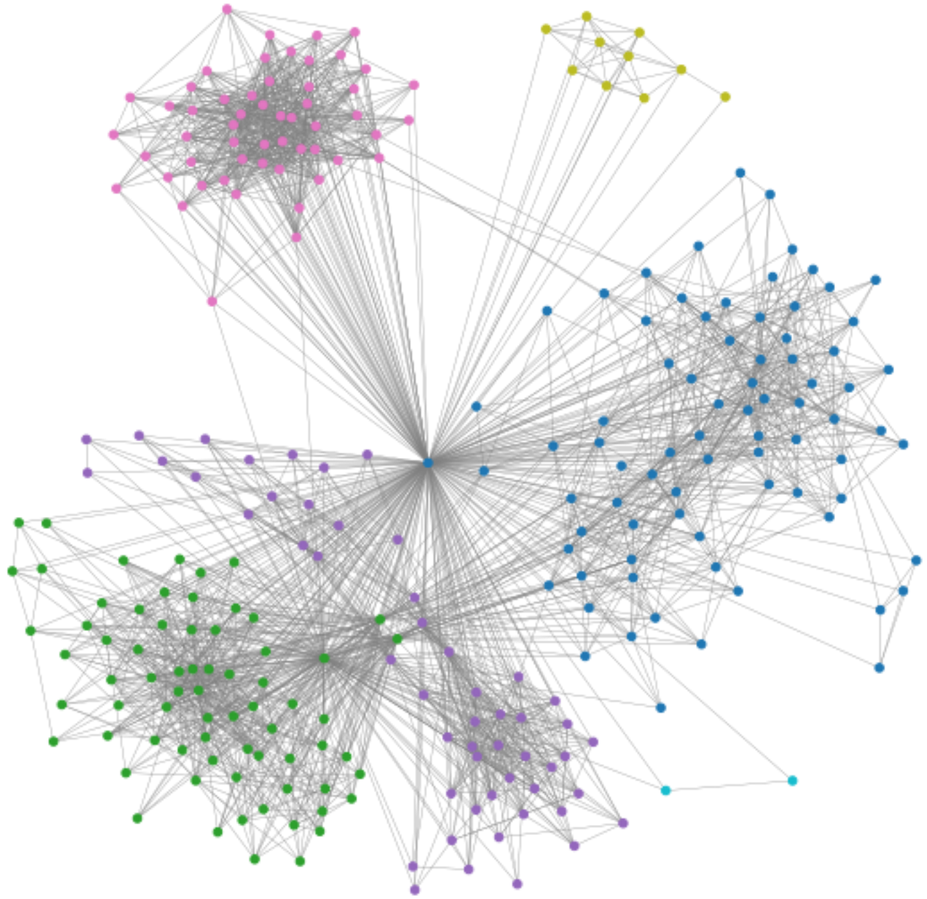
CS3102 EDA

Índice



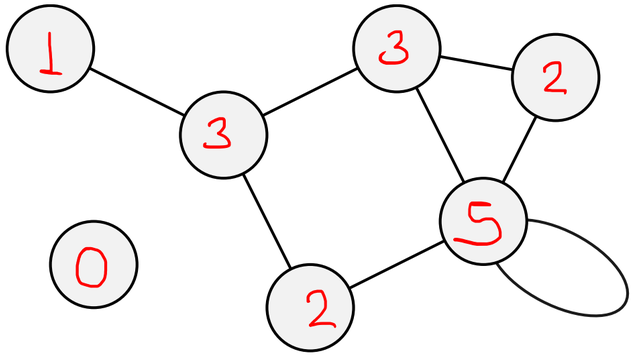
1. Graph

Gra*ph*

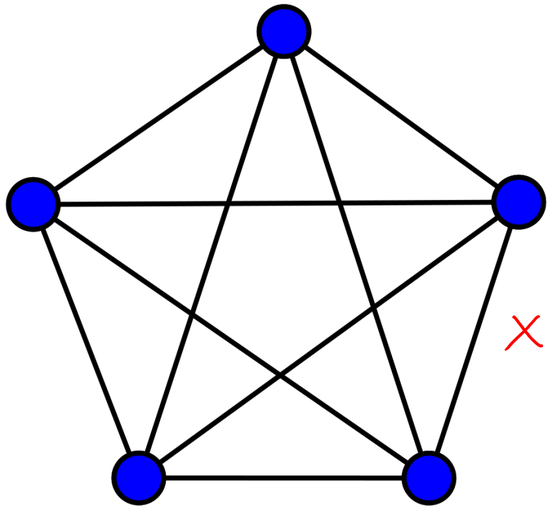


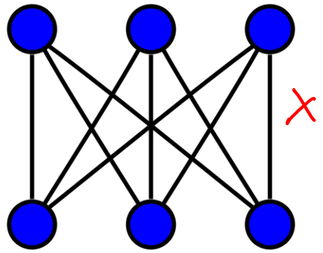
Graph

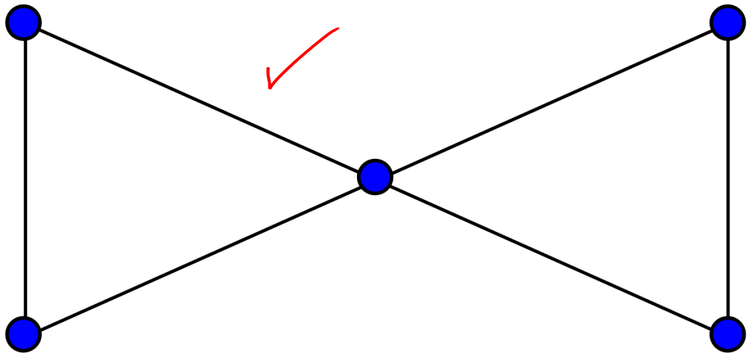
Grado del grafo

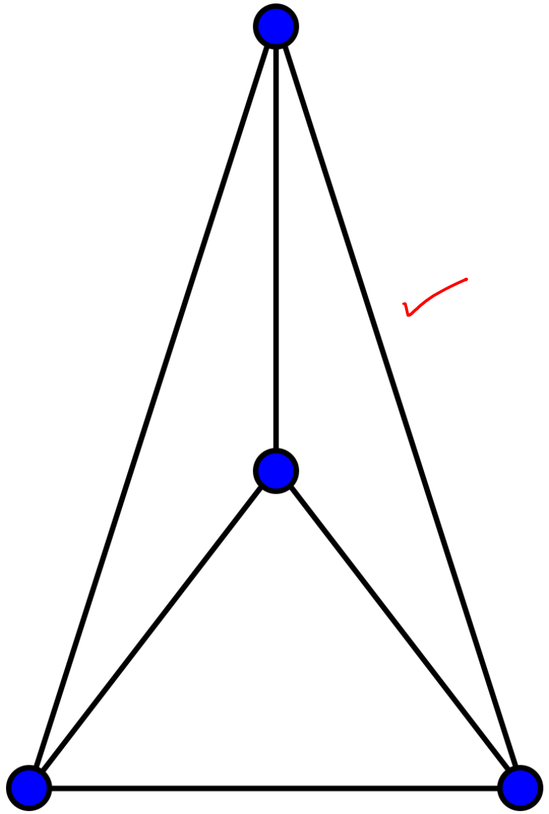


Planar *graph*

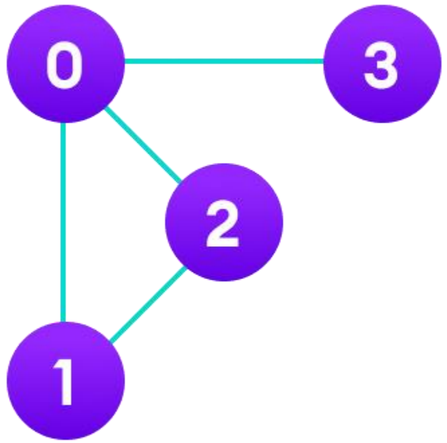








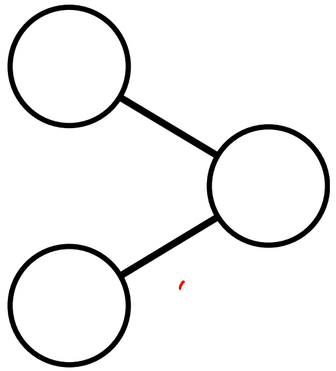
Adjacency *Matrix*



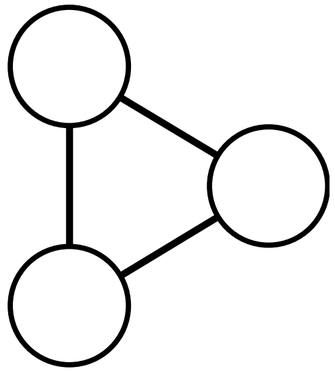
j \ i	0	1	2	3
0	0	1	1	1
1	1	0	1	0
2	1	1	0	0
3	1	0	0	0

Triplet *node*

Un **triplet** es un conjunto de tres nodos conectados por dos (**open triplet**) o tres (**closed triplet**) aristas.



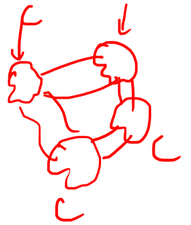
Open triplet

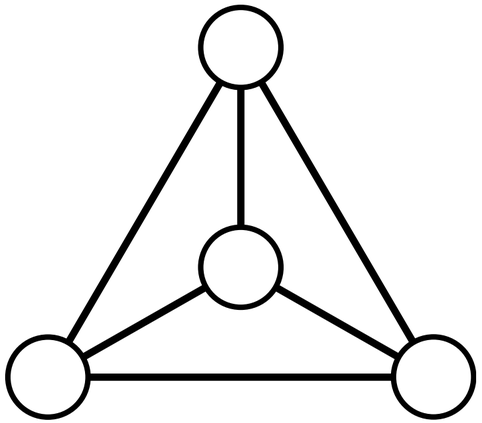


Closed triplet

Clique

Un **clique** es un conjunto (o subconjunto) de nodos completamente conectados.





Coeficiente de *agrupamiento local*

Cuantifica que tan cerca están los vecinos de un nodo i de ser un **clique**.

$$C_i = \frac{\text{Número pares de vecinos de } i \text{ que están conectados}}{\text{Número total de pares de vecinos de } i}$$

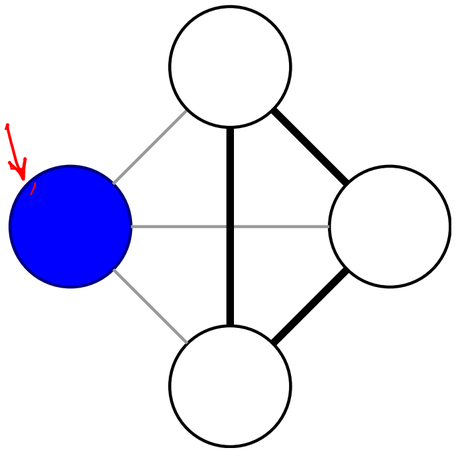
$$C_i = \frac{1}{k_i(k_i - 1)} \sum_{j,k} A_{ij} A_{jk} A_{ki}$$

$$k_i = \sum_j A_{ij}$$

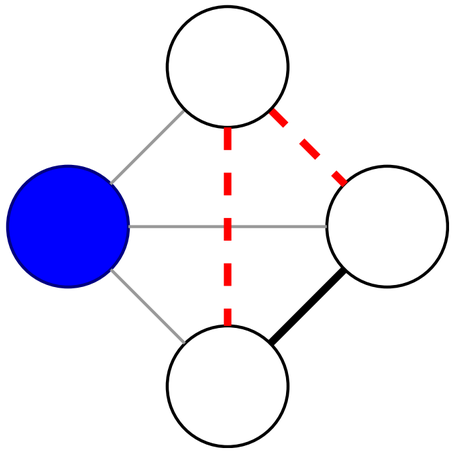
Si $k = 0$ o 1, $C_i = 0$

Coeficiente de *agrupamiento local*

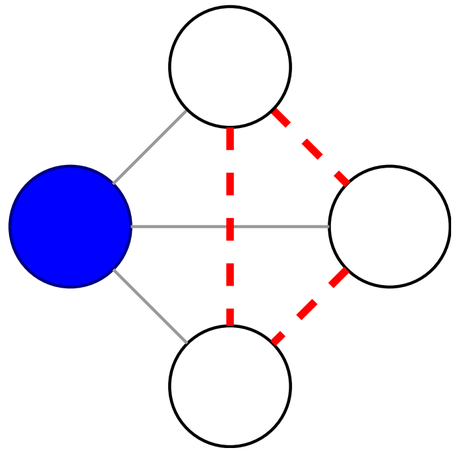
Número de posibles pares de nodos: $\binom{3}{2} = \frac{3(3-1)}{2} = 3$



$C_i = 1 = \frac{3}{3}$



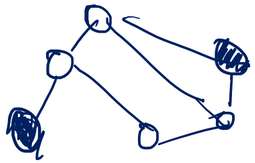
$C_i = \frac{1}{3}$

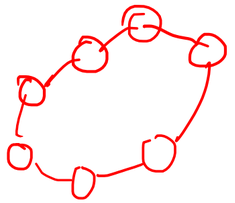


$C_i = 0$

Coeficiente de *agrupamiento global*

Percolación





$$C = \frac{\text{Número de } closed\ triplets}{\text{Número total de } triplets}$$

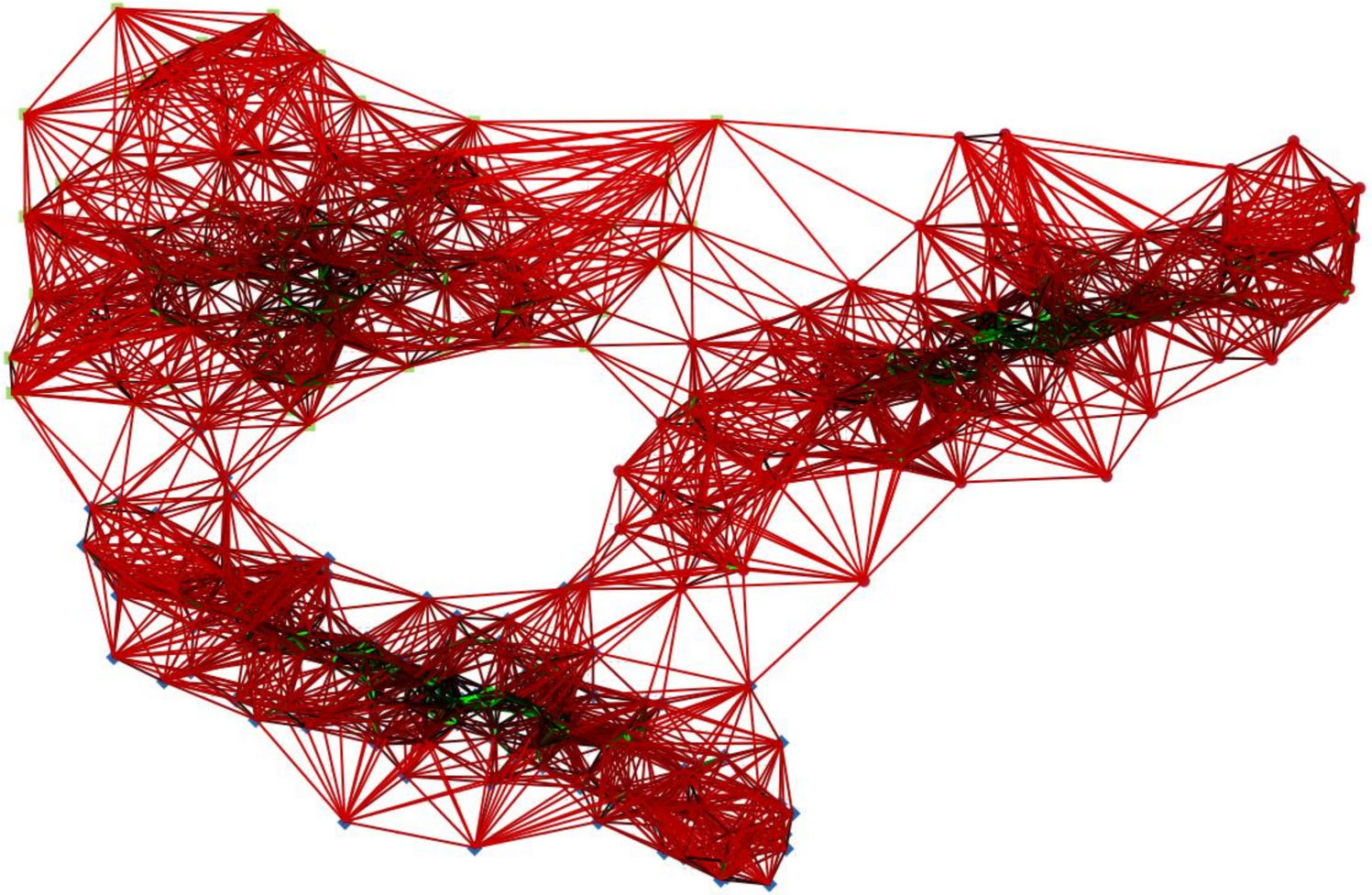
$$C = \frac{\sum_{i,j,k} A_{ij} A_{jk} A_{ki}}{\sum_i k_i (k_i - 1)}$$

$$k_i = \sum_j A_{ij}$$

Si $\sum_i k_i (k_i - 1) = 0$, $C_i = 0$

2. Nearest Neighbor Graph

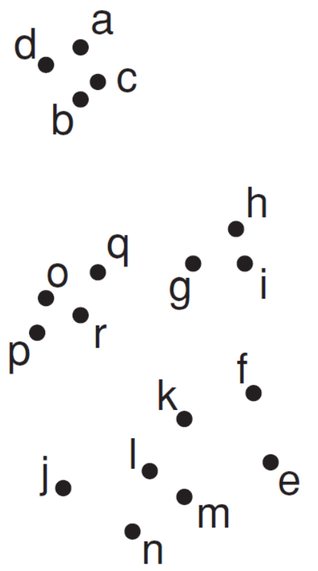
Nearest Neighbor *Graph*



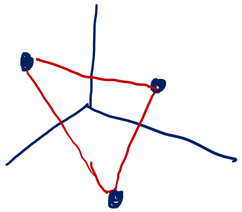
Laurens van der Maaten and Geoffrey Hinton (2008) "Visualizing Data using t-SNE".
Journal of machine learning research, vol. 9, no 11.

Nearest Neighbor *Graph*

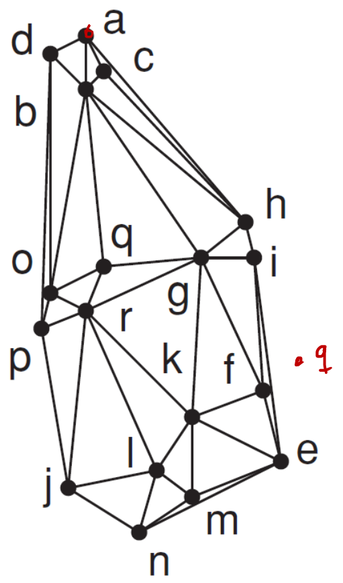
Object	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
a	0	75	56	56	660	559	352	346	393	638	558	621	667	704	366	417	326	387
b	75	0	35	71	593	493	288	293	336	563	486	547	594	629	292	343	251	313
c	56	35	0	79	604	503	296	292	338	590	503	567	613	652	321	373	275	338
d	56	71	79	0	660	562	358	363	407	613	550	606	656	686	337	388	309	366
e	660	593	604	660	0	103	309	341	290	302	140	175	135	224	403	386	372	348
f	559	493	503	562	103	0	207	239	188	307	107	188	180	266	330	325	285	274
g	352	288	296	358	309	207	0	80	75	375	225	306	338	397	218	246	138	179
h	346	293	292	363	341	239	80	0	52	451	285	372	395	462	293	324	210	257
i	393	336	338	407	290	188	75	52	0	418	241	330	349	420	292	316	213	249
j	638	563	590	613	302	307	375	451	418	0	202	127	175	118	276	228	316	251
k	558	486	503	550	140	107	225	285	241	202	0	90	112	179	266	247	247	212
l	621	547	567	606	175	188	306	372	330	127	90	0	63	91	292	258	297	246
m	667	594	613	656	135	180	338	395	349	175	112	63	0	90	350	319	348	302
n	704	629	652	686	224	266	397	462	420	118	179	91	90	0	360	319	378	321
o	366	292	321	337	403	330	218	293	292	276	266	292	350	360	0	52	84	56
p	417	343	373	388	386	325	246	324	316	228	247	258	319	319	52	0	124	67
q	326	251	275	309	372	285	138	210	213	316	247	297	348	378	84	124	0	67
r	387	313	338	366	348	274	179	257	249	251	212	246	302	321	56	67	67	0



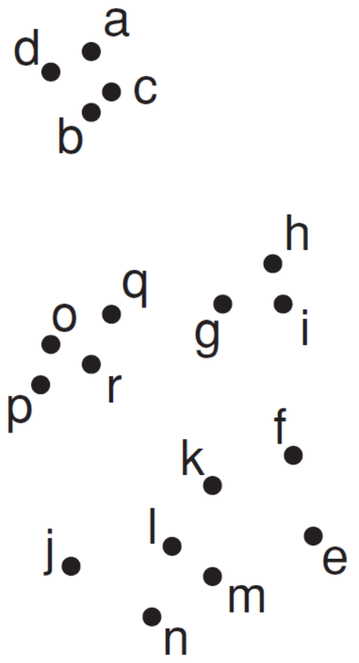
Nearest Neighbor *Graph*



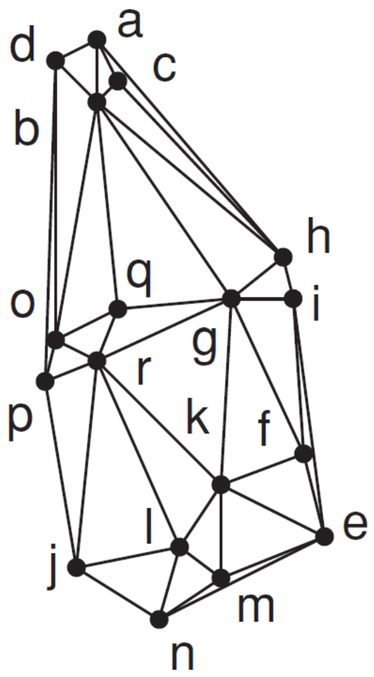
Object	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
a	0	75	56	56	660	559	352	346	393	638	558	621	667	704	366	417	326	387
b	75	0	35	71	593	493	288	293	336	563	486	547	594	629	292	343	251	313
c	56	35	0	79	604	503	296	292	338	590	503	567	613	652	321	373	275	338
d	56	71	79	0	660	562	358	363	407	613	550	606	656	686	337	388	309	366
e	660	593	604	660	0	103	309	341	290	302	140	175	135	224	403	386	372	348
f	559	493	503	562	103	0	207	239	188	307	107	188	180	266	330	325	285	274
g	352	288	296	358	309	207	0	80	75	375	225	306	338	397	218	246	138	179
h	346	293	292	363	341	239	80	0	52	451	285	372	395	462	293	324	210	257
i	393	336	338	407	290	188	75	52	0	418	241	330	349	420	292	316	213	249
j	638	563	590	613	302	307	375	451	418	0	202	127	175	118	276	228	316	251
k	558	486	503	550	140	107	225	285	241	202	0	90	112	179	266	247	247	212
l	621	547	567	606	175	188	306	372	330	127	90	0	63	91	292	258	297	246
m	667	594	613	656	135	180	338	395	349	175	112	63	0	90	350	319	348	302
n	704	629	652	686	224	266	397	462	420	118	179	91	90	0	360	319	378	321
o	366	292	321	337	403	330	218	293	292	276	266	292	350	360	0	52	84	56
p	417	343	373	388	386	325	246	324	316	228	247	258	319	319	52	0	124	67
q	326	251	275	309	372	285	138	210	213	316	247	297	348	378	84	124	0	67
r	387	313	338	366	348	274	179	257	249	251	212	246	302	321	56	67	67	0



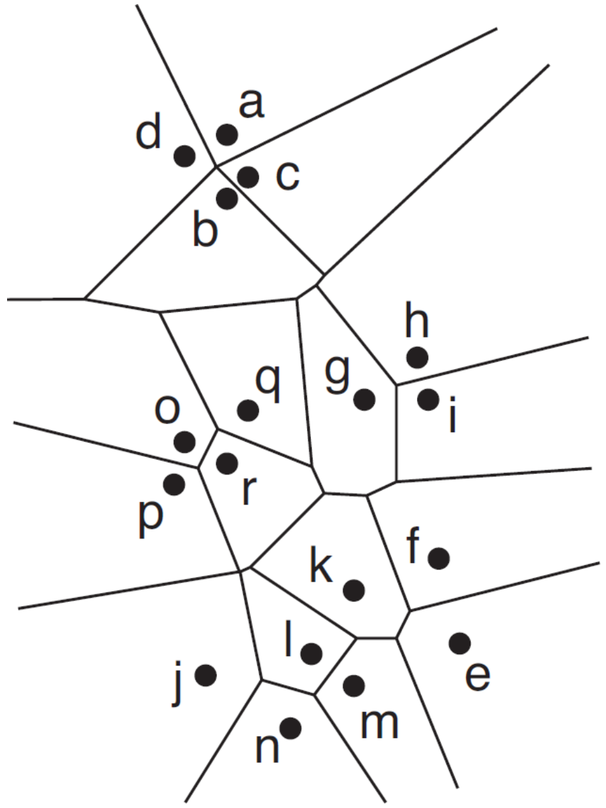
Nearest Neighbor *Graph*



Datos

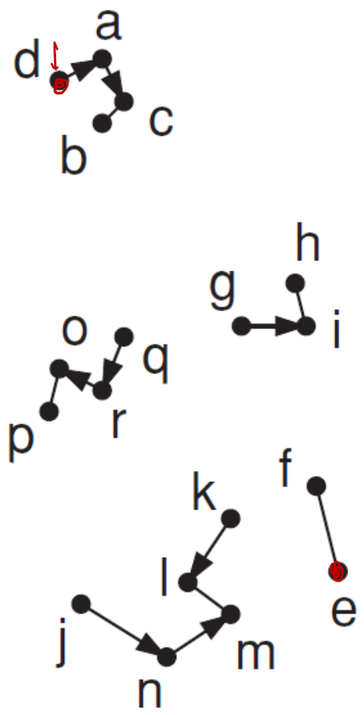


Delaunay graph



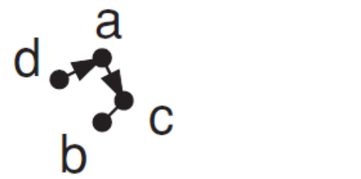
Voronoi

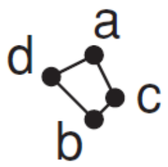
Nearest Neighbor *Graph*

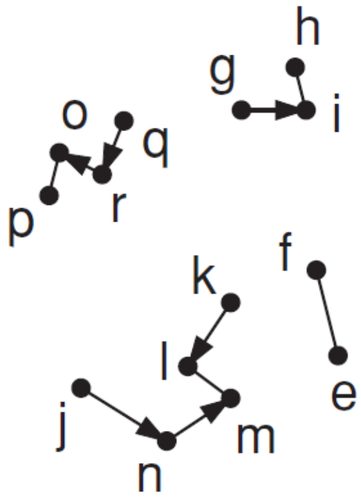


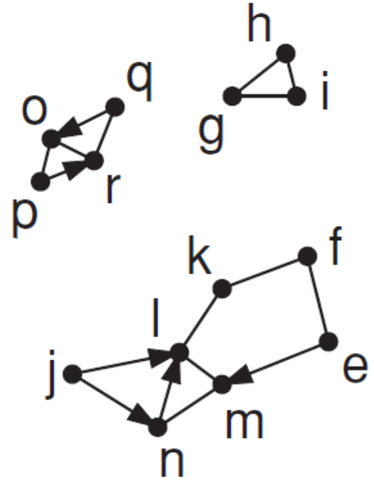
1-kNN

Nearest Neighbor *Graph*



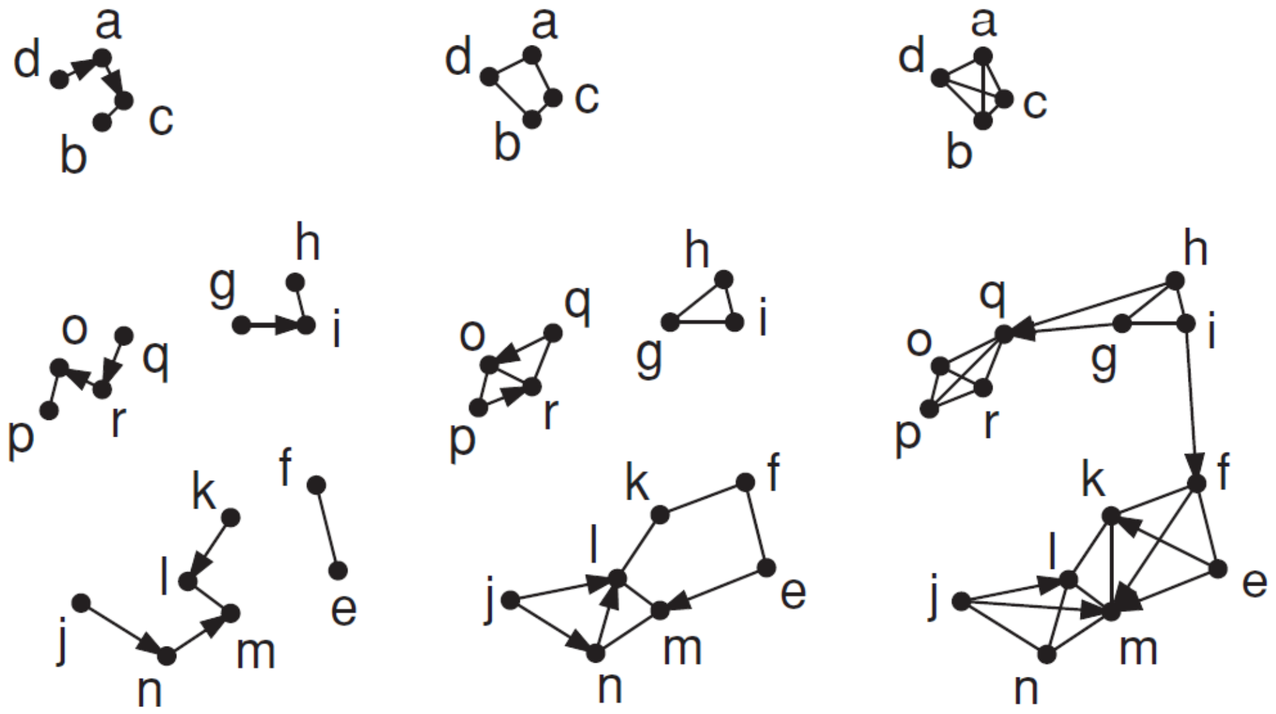






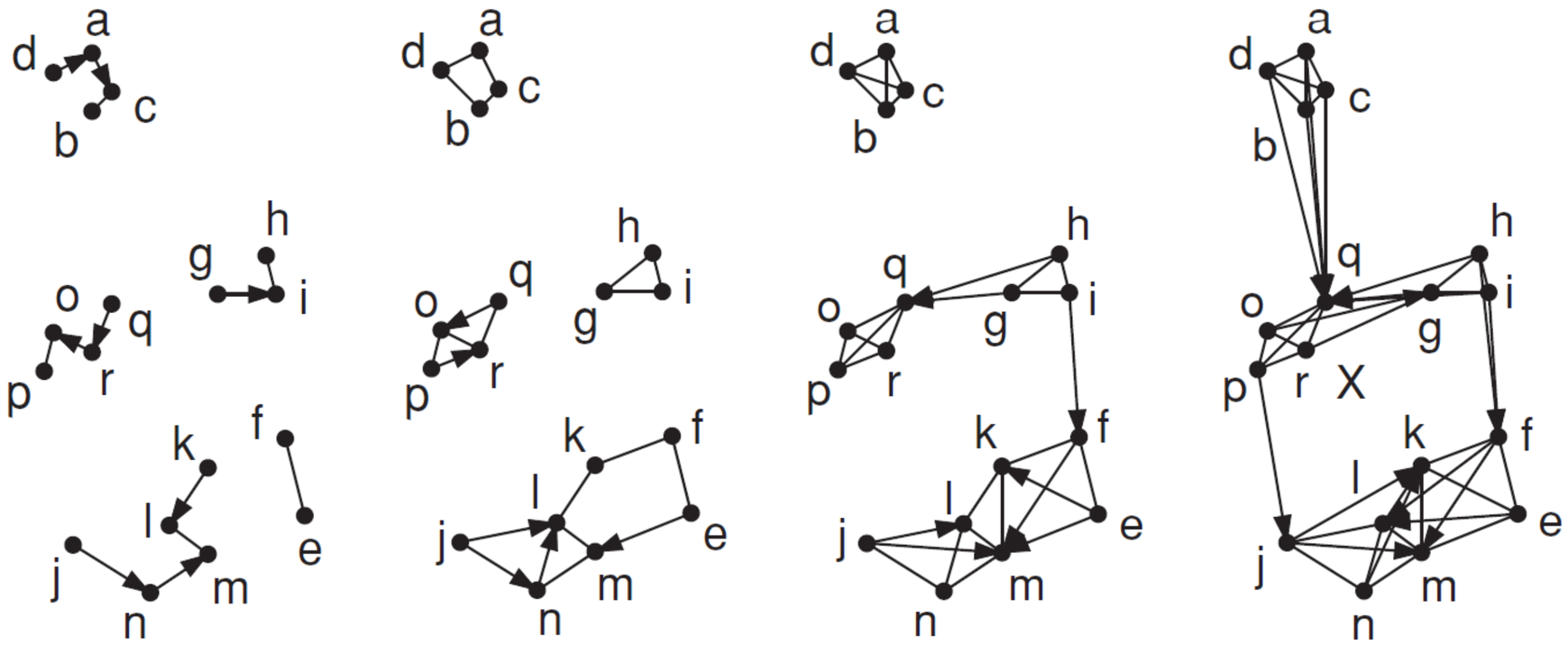
1-kNN **2-kNN**

Nearest Neighbor *Graph*



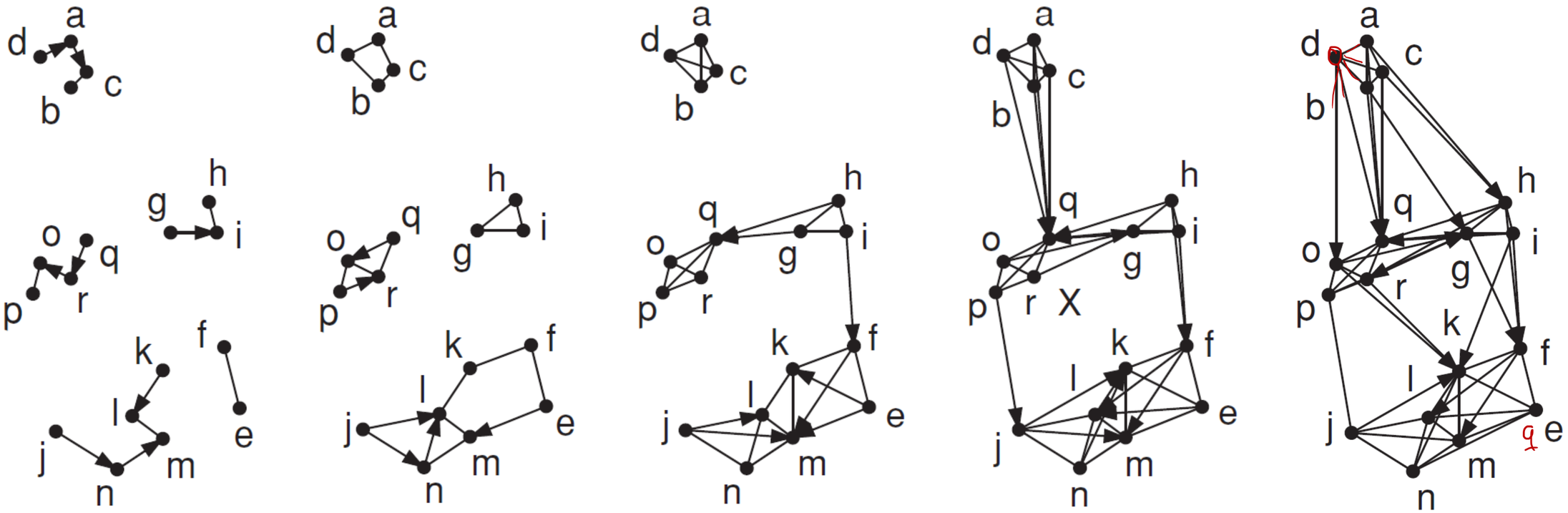
1-kNN **2-kNN** **3-kNN**

Nearest Neighbor *Graph*



1-kNN **2-kNN** **3-kNN** **4-kNN**

Nearest Neighbor *Graph*

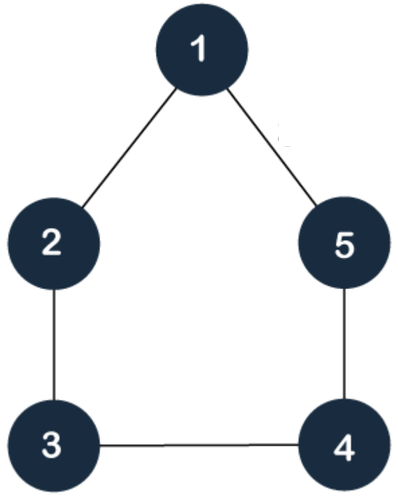


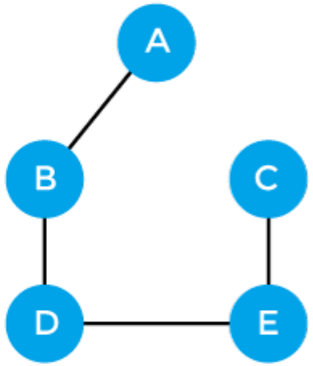
1-kNN **2-kNN** **3-kNN** **4-kNN** **5-kNN**

3. Minimum spanning tree

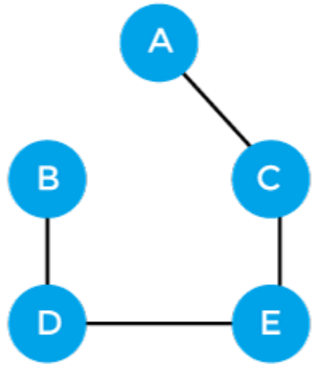
Spanning *tree*

Dado un *connected undirected graph* $G = (V, E)$, un *spanning tree* es un árbol $T = (V, E')$ con $E' \subseteq E$.

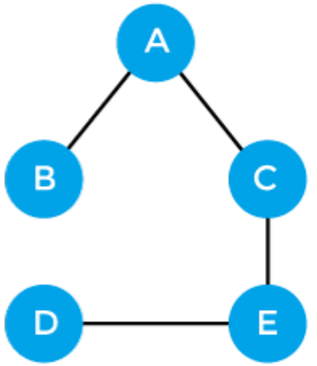




Spanning tree 1



Spanning tree 2

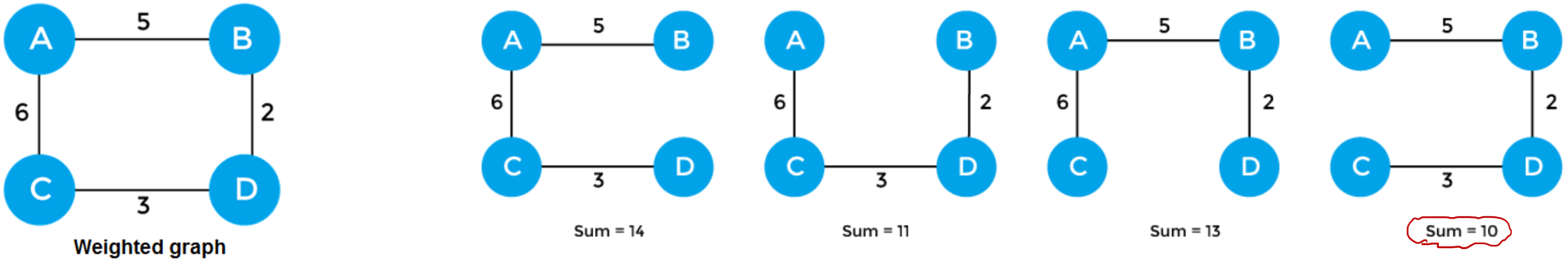


Spanning tree 3

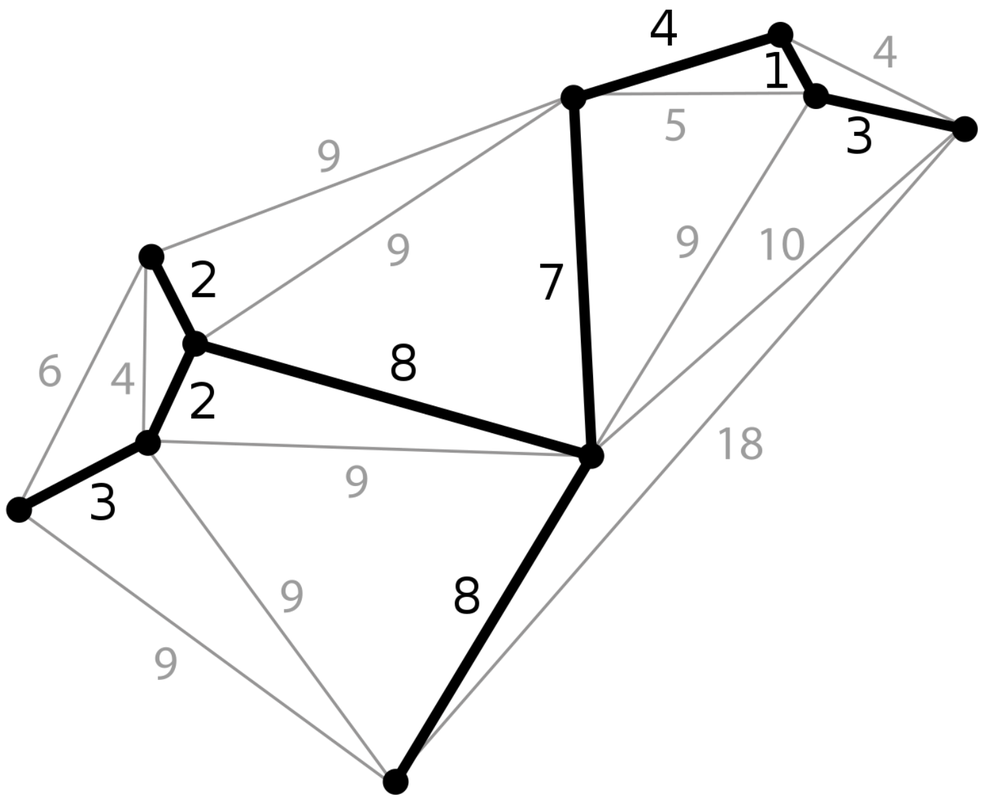
Minimum *spanning tree*

Dado un *connected, undirected weighted graph* $G = (V, E, \omega)$, el *mínimum (weight) spanning tree (MST)* es un *spanning tree* de mínimo *weight*, donde el *weight* de un árbol es definidos como:

$$\omega(T) = \sum_{e \in E(T)} \omega(e)$$



Minimum *spanning tree*

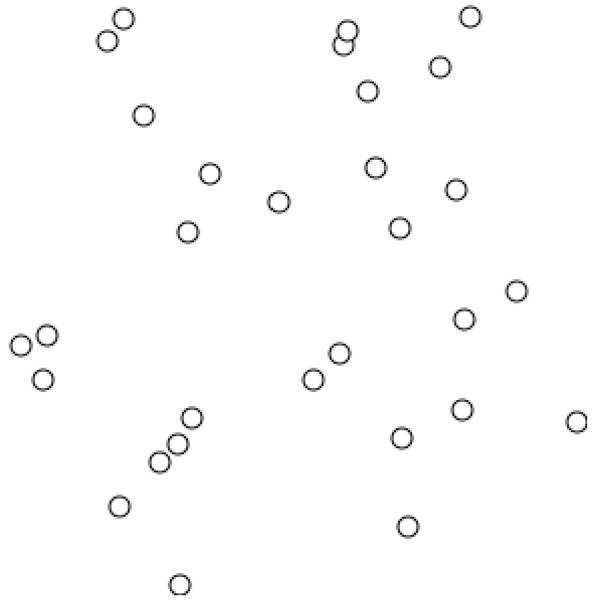


Kruskal's *algorithm*

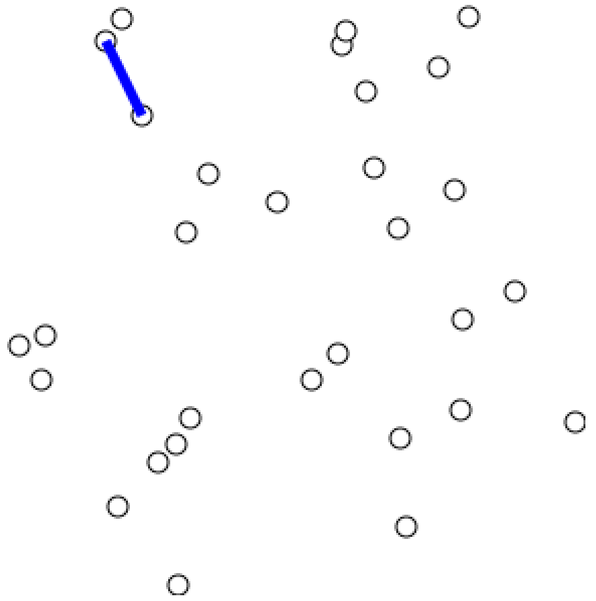
1. Ordena todos los *edges* de menor a mayor.

2. Agregue el *edge* más pequeño al *spanning tree* final. Si agregar el *edge* crea un ciclo, pase al siguiente *edge*.

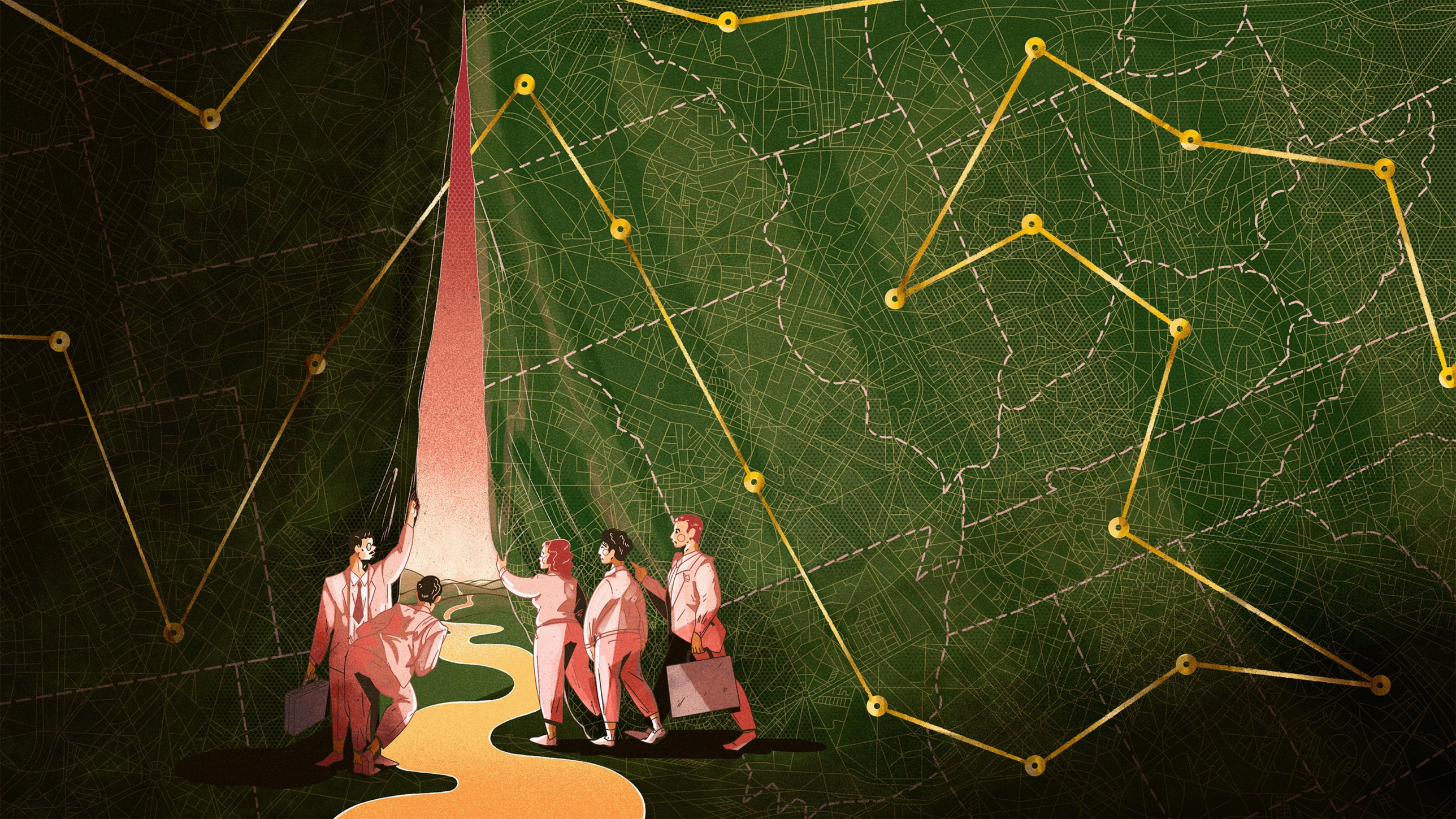
3. Siga agregando todos los *edges* hasta que todos los nodos formen parte del *tree*.



Prim's *algorithm*

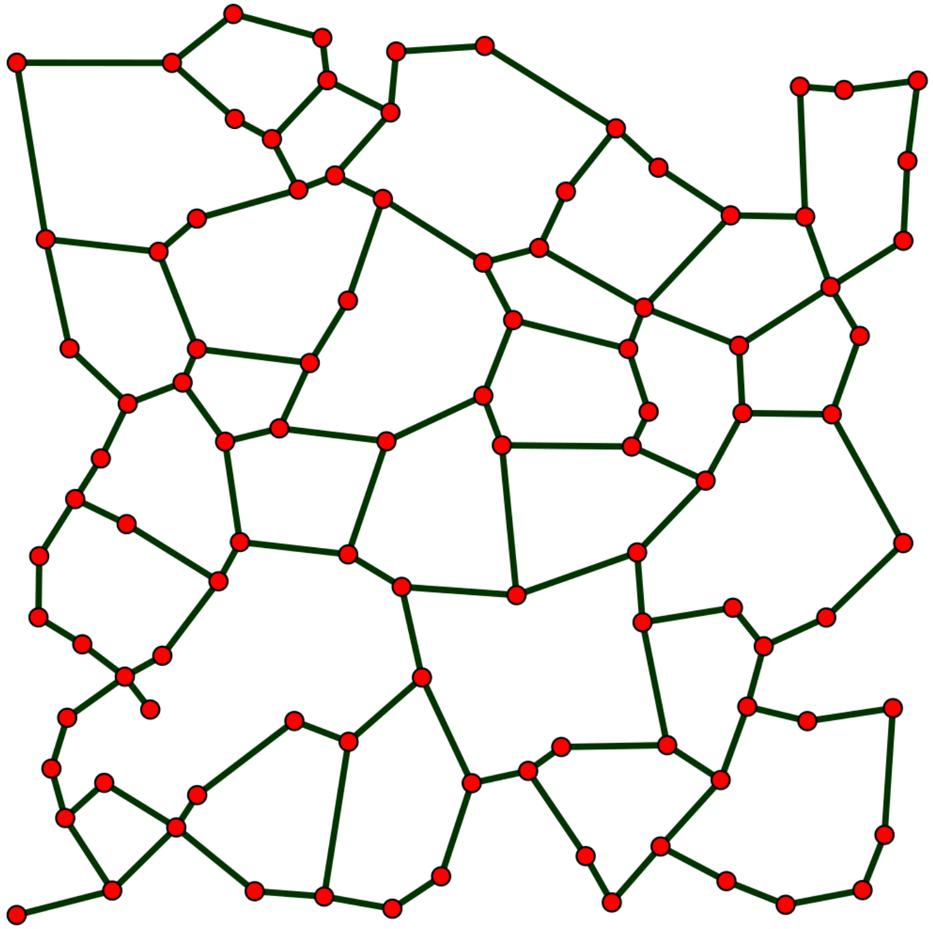


1. Elija y visite un nodo arbitrario.
2. De todos los bordes visitados, elija el más pequeño y agregue el *edge* al árbol.
3. Siga agregando *edges*, hasta que se visiten todos los nodos.



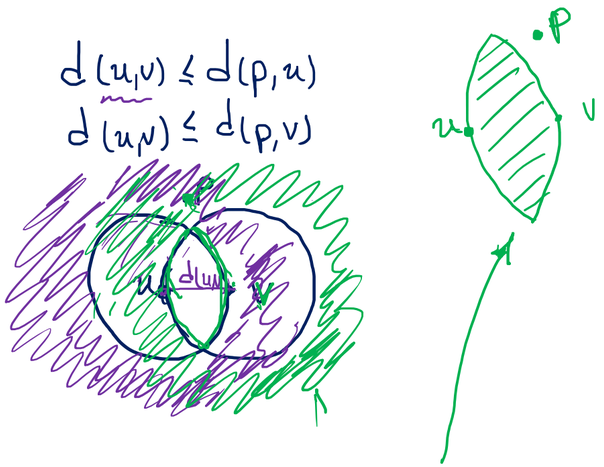
4. Relative neighborhood graph

Relative neighborhood *graph (RNG)*



Convexo ≃ Navegable

Planar



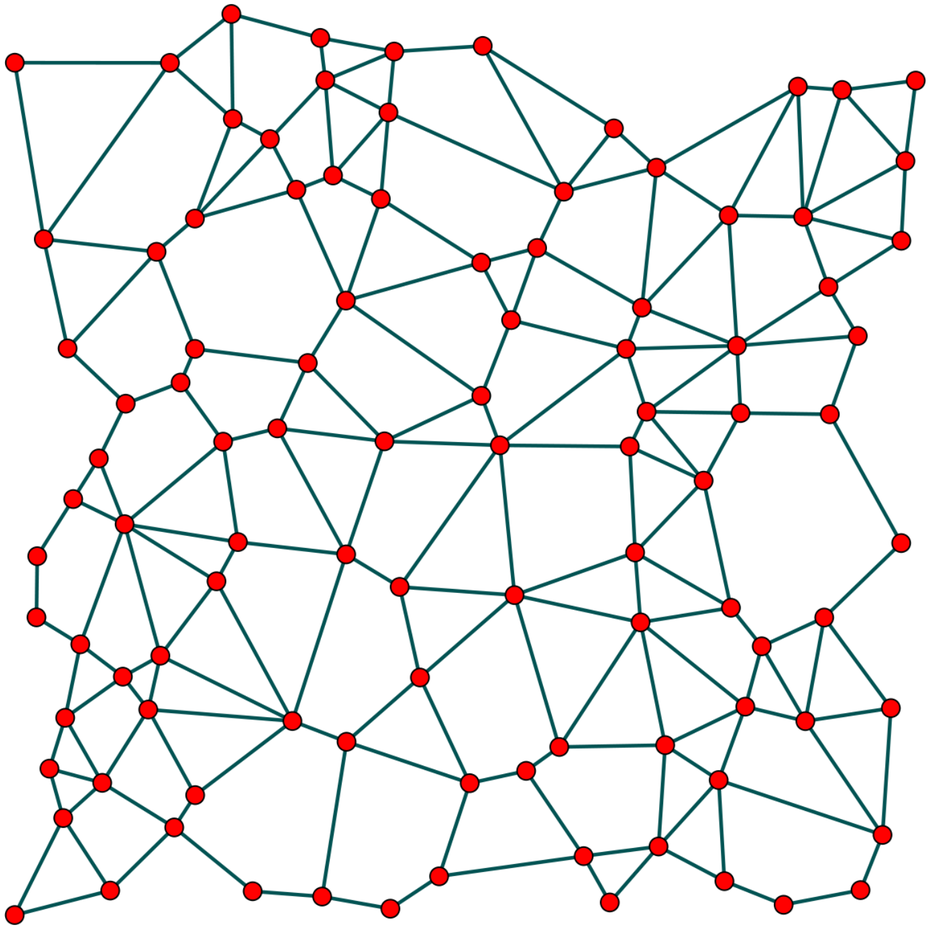
$$d(u,v) \leq \max\{d(p,u), d(p,v)\}$$

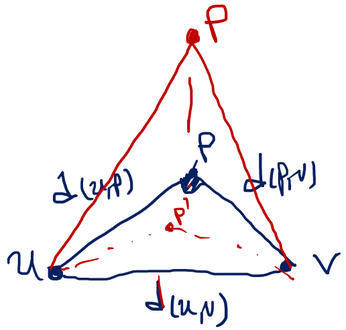
$$\ell(n_1, n_2) \leq 2\, d_2(n_1, n_2)$$

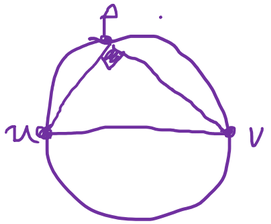
Godfried T. Toussaint (1980) "The relative neighborhood graph of a finite planar set".
Pattern recognition, vol. 12, no 4, p. 261-268.

5. Grafos graph

Gabriel *graph (GG)*





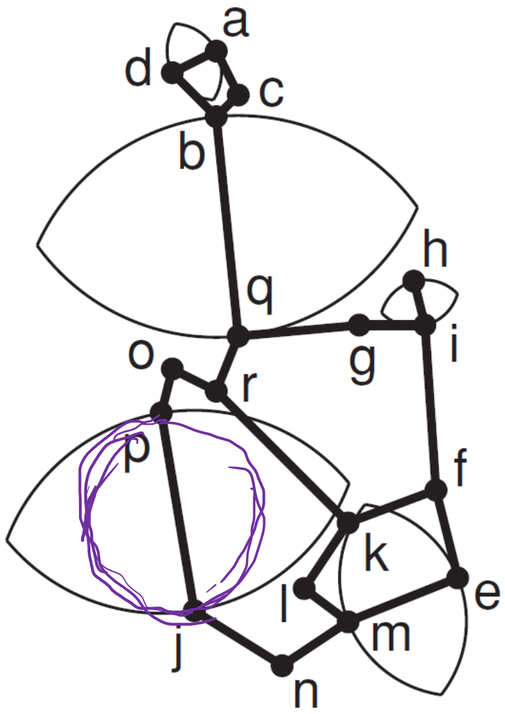


$$d(u,v)^2 \leq d(u,p)^2 + d(p,v)^2$$

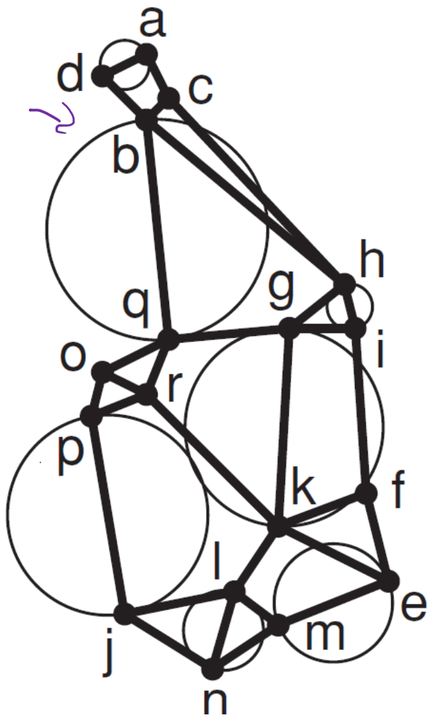
Plano
Convexo

K. Ruben Gabriel and Robert R. Sokal (1969) "A new statistical approach to geographic variation analysis".
Systematic Biology, Volume 18, Issue 3, Pages 259–278

Gabriel *graph*

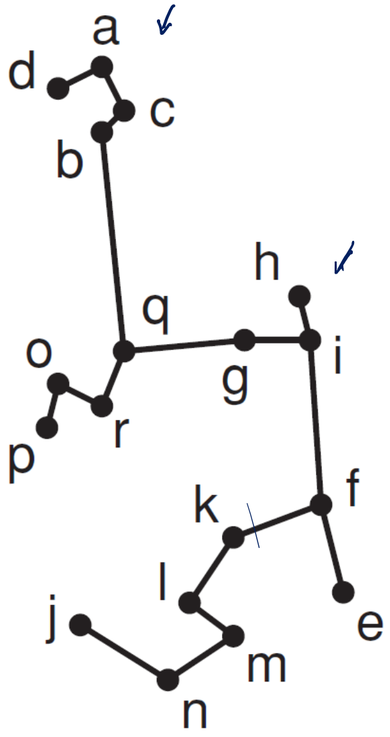


Relative neighborhood graph (RNG)

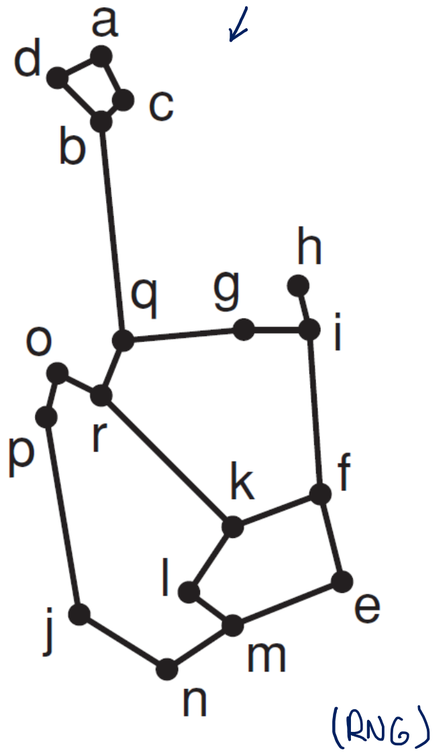


Gabriel graph (GG)

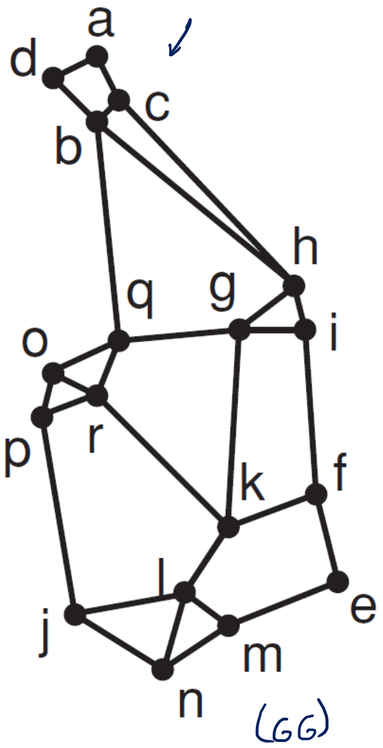
Gabriel *graph*



Minimal spanning tree

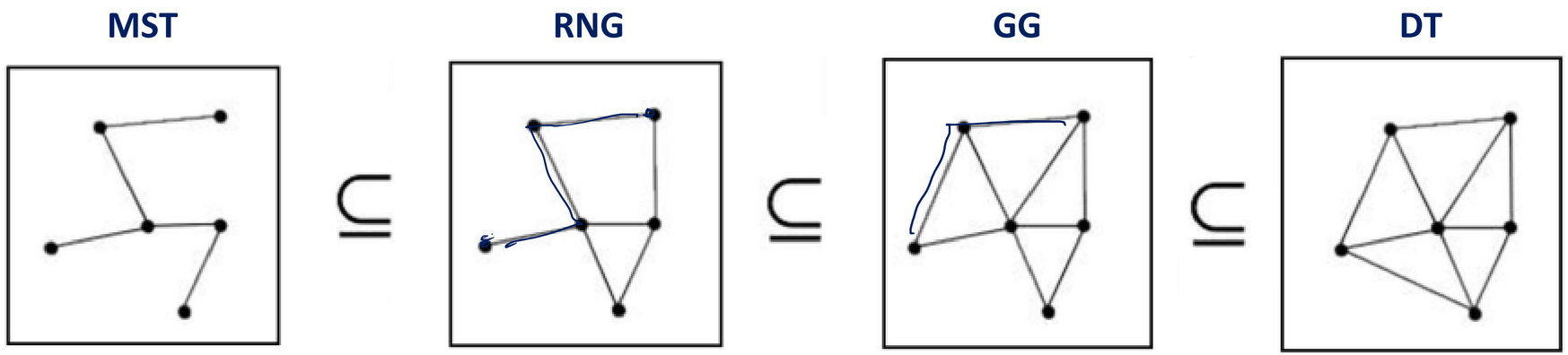


Relative neighborhood graph



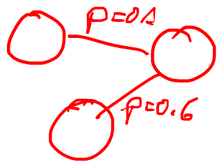
Gabriel graph

Gabriel *graph*



6. Grafos aleatorios

Grafo *aleatorio*



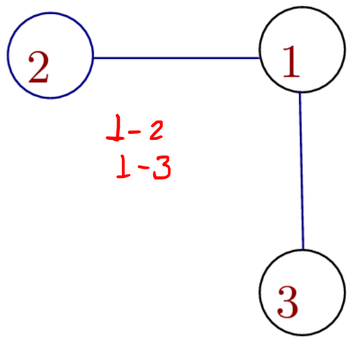
Uniform random graph

n : nodos

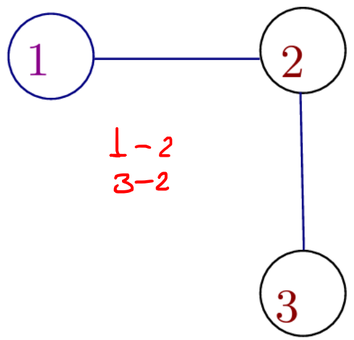
Grafos con exactamente m aristas

$$\frac{1}{\binom{\binom{n}{2}}{m}}$$

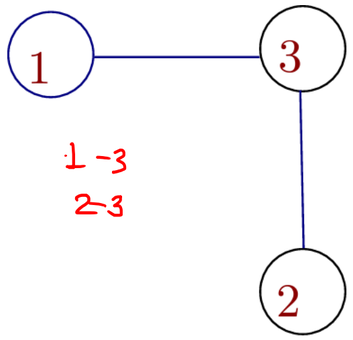
m = 2



Probability $\frac{1}{3}$



Probability $\frac{1}{3}$



Probability $\frac{1}{3}$

Grafo *aleatorio*

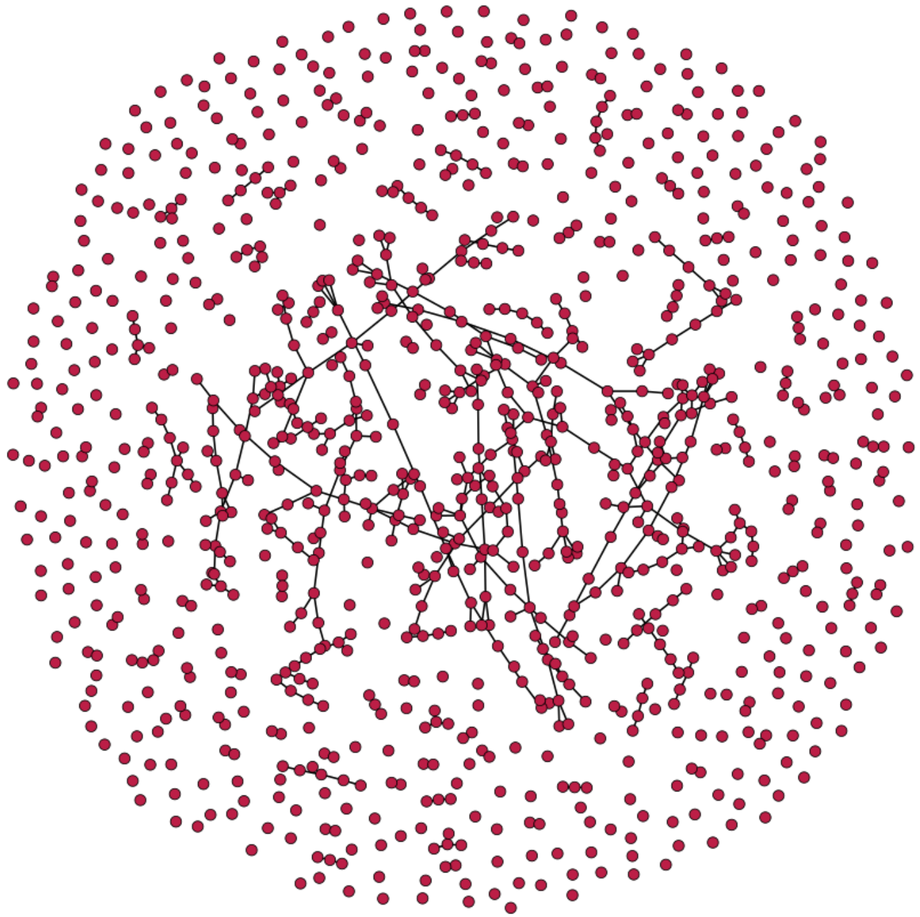
Percolación

Modelo de Erdős–Rényi $G(n, m)$

Random binomial graphs

Si cada par de nodos tiene probabilidad p de estar conectados, entonces, la probabilidad de obtener un grafo de n nodos y m aristas:

$$p^m (1-p)^{\binom{n}{2}-m}$$



UTEC

UNIVERSIDAD DE INGENIERÍA
Y TECNOLOGÍA