

①

D M A

Scribē

1 2 3 ... N \rightarrow Indicas una lista con N y otra lista con N

1
2
3
:
N

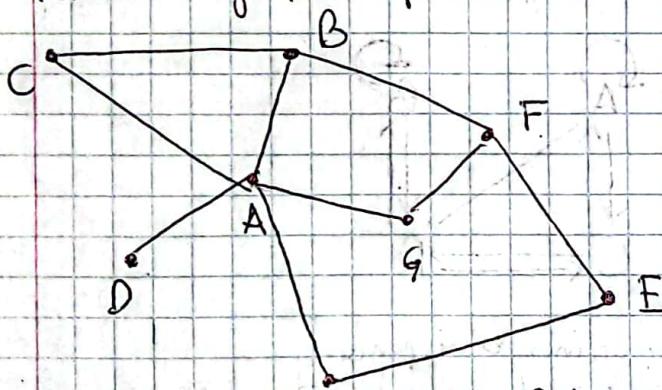
	Listas 1	Listas 2
1	1	1
2	2	2
3	3	3
:	:	:
N	N	N

\rightarrow Ahora bajo una función al azar, generas una probabilidad o condición en la que si se cumple, entonces forma una unión o interacción.

- * Esto puede ser un número fijo $[0, 1]$, en el que $x \sim f(x) | x \in [0, 1]$
- \Rightarrow es más probable que suceda, osca da si el número al caer en $\{P \Rightarrow$ sucede interacción y si $> P$ entonces no sucede.

\rightarrow Ahora hay que saber qué tan conectados están entre sí los nodos.

Tomemos el ejemplo de por acá



	A	B	C	D	E	F	G	H
A	0	1	1	1	0	0	1	1
B	1	0	1	0	0	1	0	0
C	1	1	0	0	0	0	0	0
D	1	0	0	0	0	0	0	0
E	0	0	0	0	0	1	0	1
F	0	1	0	0	1	0	1	0
G	1	0	0	0	0	1	0	0
H	1	0	0	0	1	0	0	0

Podemos considerar solo un lado, ahorrando para hacer cálculos de clustering. Considera cuántos nodos tiene tiene conectados el nodo I y cuántos entre ellos están conectados.

$$\Lambda = \text{No. vecinos entre vecinos del nodo } I$$

$$k = \text{Vecinos que tiene conectados con el nodo } I$$

$$C_I = \text{Coeficiente de clustering}$$

$$C_I = \frac{2\Lambda}{k(k-1)}$$

$$\begin{array}{ll} C_A = 0.1 & C_F = 0 \\ C_B = 0 & C_F = 0 \\ C_C = 0 & C_G = 0 \\ C_D = 0 & C_H = 0 \end{array}$$

A	0	1	1	1	0	0	1	1
B	0	1	0	0	1	0	0	0
C	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0
E	0	1	0	0	1	0	1	0
F	0	1	0	0	0	1	0	0
G	1	0	0	0	0	1	0	0
H	1	0	0	0	1	0	0	0

¿Forma de algo analizar las conexiones entre vecinos?

Si $A \rightarrow B \wedge B \rightarrow C \wedge A \rightarrow C \Rightarrow$ triangle or $n = 1$

En secuencia si analiza 2 a la vez, como verás?

— Pero en código no se puede, por ende creamos comparaciones.

- Tenemos que para todos aquellos nodos que sean 1, agrupados para comparar aquello nodes que tienen pareja con su mismo, cuentan.

- Se hacen comparaciones de 3 en 3

toma A, B, C :

$\text{if } (A, B) = 1 \text{ and } (B, C) = 1 \text{ and } (A, C) = 1 \Rightarrow$
 \Rightarrow triangle or están conectados

Otra se hará una serie de combinaciones, de tal manera que se comparan entre todos, las combinaciones serán basadas en cada uno no existente y que forme parejas pos.

Segundo código función (N, P)

Lista 1 = []

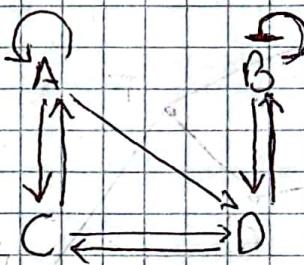
Lista 2 = []

rango (N)

En este momento se me ocurrió pensar que si se genera una matriz aleatoria entonces tomaremos el ejemplo

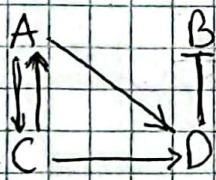
	A	B	C	D
A	1	0	1	1
B	0	1	0	1
C	1	0	0	1
D	0	1	1	0

Modelo visual

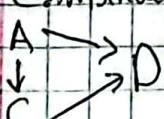


Solo hablaremos de si hay conexión entre vecinos o responder
 ¿Están conectados? ¿Sí o No? Mas no especificar si dirigida o en una o
 otra dirección.

Así que si por ejemplo está esto



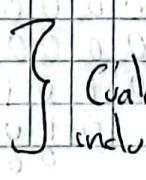
Combinaciones



	A	B	C	D
A	1	0	0	1
B	0	1	0	0
C	1	0	0	1
D	0	1	0	0

Contemplando que va de $A \rightarrow N$

Así:
 $B \rightarrow N$
 $C \rightarrow N$
 $D \rightarrow N$



Cualquier de los 2 cuenta como conexión entre vecinos
 incluso si tienen $\overset{\rightarrow}{2}$ no importa, solo contempla si
 ¡Sí o No!

Seudo código
funcion (N, p)

$\text{Lista1} = [] \quad \} \text{range}(N)$
 $\text{Lista2} = []$

- Crear dataframe (N)

for i in range(N):
 for j in range(N):

 random() \rightarrow que de un numero entre [0, 1]

 if random() < p:

 Nodo[1][1] = 1, ..., Nodo[1][N] = 1

 or, Nodo[1][1] = -1, ..., Nodo[1][N] = -1

 else:

 Nodo[1][1] = 0, ..., Nodo[1][N] = 0

print(Dataframe(aleatorioN))

count autoregulations \rightarrow count(Nodo[N][N] = 1 or -1)

for a in range(N):

 for b in range(N):

 for c in range(N):

 if a \neq b \neq c:

Permutaciones Add if (Nodo[a][b] and Nodo[b][c] and Nodo[a][c]) = 1

$\bullet (a,b) \cdot (a,c) \cdot (b,c)$

triangle

$\times (b,a) \cdot (a,c) \cdot (b,c)$

clif (Nodo[b][a] and Nodo[b][c] and Nodo[a][c]) = 1

$\bullet (a,b) \cdot (c,a) \cdot (b,c)$

triangle

$\bullet (a,b) \cdot (a,c) \cdot (c,b)$

etc 3 combinaciones diferentes

~~(a,b) · (a,c) · (c,a) · (c,b)~~

triangle later 2

~~x (b,a) · (~~

$3^2 = 9$

↓ Combinaciones de num.

$$(a,b) \cdot (a,c) \cdot (b,c) - (a,b) \cdot (c,a) \cdot (b,c) = (a,b) \cdot (a,c) \cdot (b,c)$$

$$(a,b) \cdot (c,a) \cdot (b,c) = \cancel{(a,b) \cdot (c,a) \cdot (b,c)}$$

$$(b,a) \cdot (a,c) \cdot (b,c) - (b,a) \cdot (c,a) \cdot (b,c) = (b,a) \cdot (a,c) \cdot (c,b)$$

$$(b,a) \cdot (c,a) \cdot (c,b)$$

Dos de una forma se condicionea a 2 formas que de (a,b) da f₁ y de (b,a) otra (f₂) (f₁) (f₂) f₁, donde cada forma se llamaría con f₁, f₂ o f₃ de tal manera que fue la secuencia:

④

2022

D M A

Scribel

$$\begin{aligned}
 f_1 &= (a, b) \cdot (a, c) \cdot (b, c) \\
 f_2 &= (a, b) \cdot (a, c) \cdot (c, b) \\
 f_3 &= (a, b) \cdot (c, a) \cdot (b, c) \\
 f_4 &= (a, b) \cdot (c, a) \cdot (c, b) \\
 f_5 &= (b, a) \cdot (a, c) \cdot (b, c) \\
 f_6 &= (b, a) \cdot (a, c) \cdot (c, b) \\
 f_7 &= (b, a) \cdot (c, a) \cdot (b, c) \\
 f_8 &= (b, a) \cdot (c, a) \cdot (c, b)
 \end{aligned}$$

	f ₁	f ₂	f ₃	f ₄
Intento 1	1	0	0	0
Intento 2	1	1	1	1
Intento 3	0	0	0	0

Condición de un algoritmo 1, entonces se considera que es una conexión entre vecinos.

Intento 1 a, b, c
Intento 2 a, b, d

Recorrido

$$\begin{aligned}
 N^0 \text{ Lista} &= [a, b, c, d] & a & b \\
 N^1 \text{ Lista} &= [a, b, c, d] & b & c \\
 N^2 \text{ Lista} &= [a, b, c, d] & c & d
 \end{aligned}$$

Una respuesta, ya que ahora se consideran combinaciones únicas, para una matriz que al menos 1, considerando el clustering

	f ₁	f ₂	f ₃	f ₄	neighbors
a, b, c	1	0	0	0	1
a, b, d	1	1	1	1	1
a, c, d	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮

Suma de ellos.

(intentos)

* Cuando $N = b$, pasaremos a buscar a todos los intentos de a, que tienen a "b" y luego adicionarlos al Dataframe de b para hacer lo mismo de calcular el n.

* Cuando $N = c$ pasaremos a buscar en el Dataframe de a, b las que tienen C para incluirlos al Dataframe de C y continuar hasta el proceso como si hiziera con a.

Pensando bien, no es mala idea los far, por que para ser una matriz q para b, seria otra matriz q para C otra matriz q asimilaria N.

K sera calculada a partir de la matriz de Nodos aleatoria

Sumas de vecinos en el nodo

1 2 3 4 5

$$N_1 = 1 \quad N_2 = 1 \quad N_3 = 1$$

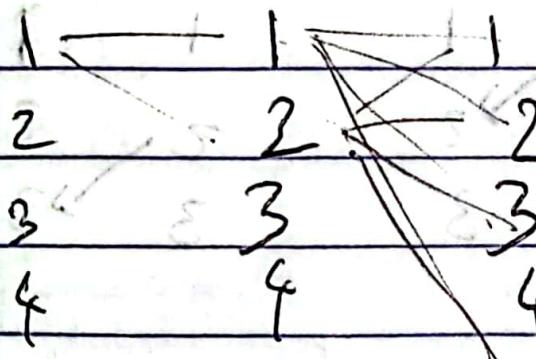
$$N_1 = N_2 \Rightarrow N_0$$

$$(N_1 = 1 \neq N_2 = 2) \Rightarrow \cancel{N_0}$$

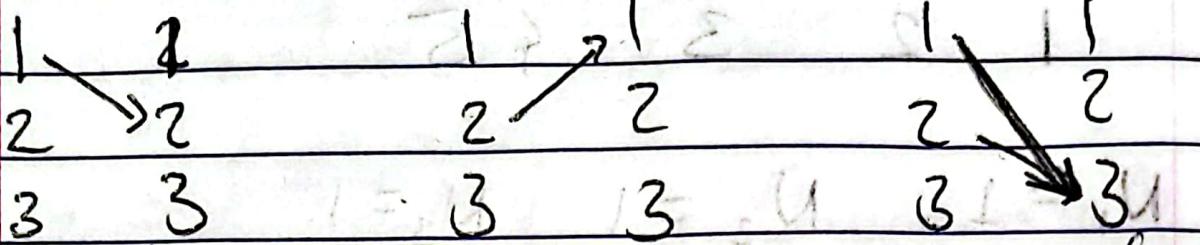
$$N \Leftrightarrow (N_1 \neq N_3)$$

$$(N_2 \neq N_3)$$

$$N_1 \neq N_2 \neq N_3$$

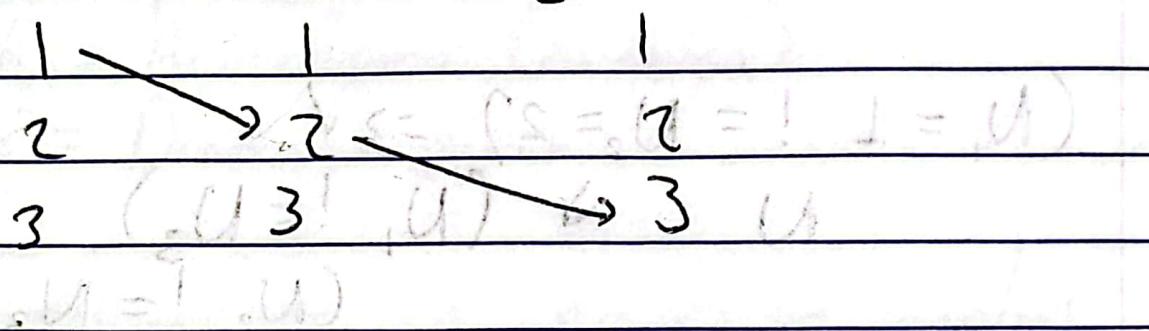


$$N_1 \neq N_2 \Rightarrow N_2 \neq N_3 \Rightarrow N_1 \neq N_3$$

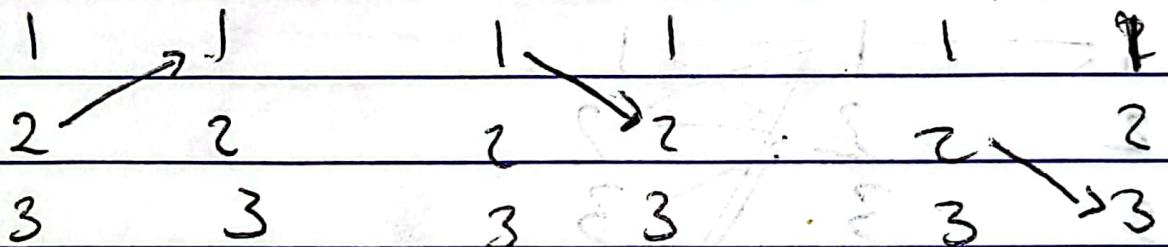


Siguendo la secuencia anterior

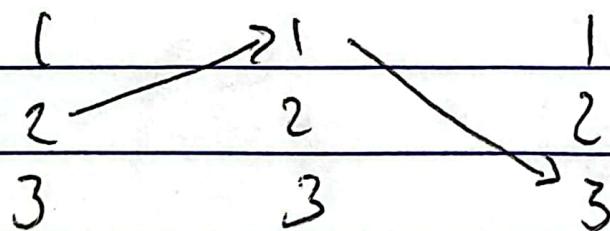
$$N_1 \neq N_2 \neq N_3$$



$$N_1 \neq N_2 \Rightarrow N_2 \neq N_3 \Rightarrow N_1 \neq N_3$$

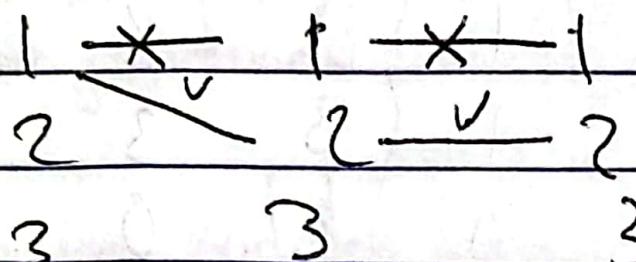


$$N_1 \neq N_2 \neq N_3 \quad \text{Secuencia 3.f}$$



Primera instrucción

$$N_1 \neq N_2 \neq N_3$$



Ahora para el caso en el que sucede

	1	2	3	4	
1	0	1	0	1	
2	1	0	0	1	
3	1	1	1	1	
4	0	0	0	0	

~~$E_{1,2} = 0$~~ $\overset{1}{[1,2]}$ $\overset{1}{[1,3]}$ $[2,3]$
 $[N_1, N_2]$, $[N_1, N_3]$, $[N_2, N_3]$

1, 2	1, 3	2, 3
1, 2	1, 4	2, 4
3, 1	3, 4	1, 4
3, 1	3, 2	2, 3

El algoritmo da a seguir para calcular el clustering de cada nodo por los criterios y arribar a un resultado, es para asignar un valor a cada nodo.

	1	2	3	4	5	
1	1	1	0	-1	0	{ 4 } rep 4
2	0	-1	0	0	-1	{ 4 } 3
3	1	1	1	-1	1	{ 5 } 3
4	0	0	1	0	0	{ 2 } 0
5	0	1	-1	0	1	{ 3 } 1
	4	4	5	2		Combinaciones
	1	2	3	4	5	$5C3 = 10$
	0	1	2	3	4	
	5	6	7	8	9	
	10	11	12	13	14	
	15	16	17	18	19	
	20	21	22	23	24	
	25	26	27	28	29	
	30	31	32	33	34	
	35	36	37	38	39	
	40	41	42	43	44	
	45	46	47	48	49	
	50	51	52	53	54	
	55	56	57	58	59	
	60	61	62	63	64	
	65	66	67	68	69	
	70	71	72	73	74	
	75	76	77	78	79	
	80	81	82	83	84	
	85	86	87	88	89	
	90	91	92	93	94	
	95	96	97	98	99	
	100	101	102	103	104	
	105	106	107	108	109	
	110	111	112	113	114	
	115	116	117	118	119	
	120	121	122	123	124	
	125	126	127	128	129	
	130	131	132	133	134	
	135	136	137	138	139	
	140	141	142	143	144	
	145	146	147	148	149	
	150	151	152	153	154	
	155	156	157	158	159	
	160	161	162	163	164	
	165	166	167	168	169	
	170	171	172	173	174	
	175	176	177	178	179	
	180	181	182	183	184	
	185	186	187	188	189	
	190	191	192	193	194	
	195	196	197	198	199	
	200	201	202	203	204	
	205	206	207	208	209	
	210	211	212	213	214	
	215	216	217	218	219	
	220	221	222	223	224	
	225	226	227	228	229	
	230	231	232	233	234	
	235	236	237	238	239	
	240	241	242	243	244	
	245	246	247	248	249	
	250	251	252	253	254	
	255	256	257	258	259	
	260	261	262	263	264	
	265	266	267	268	269	
	270	271	272	273	274	
	275	276	277	278	279	
	280	281	282	283	284	
	285	286	287	288	289	
	290	291	292	293	294	
	295	296	297	298	299	
	300	301	302	303	304	
	305	306	307	308	309	
	310	311	312	313	314	
	315	316	317	318	319	
	320	321	322	323	324	
	325	326	327	328	329	
	330	331	332	333	334	
	335	336	337	338	339	
	340	341	342	343	344	
	345	346	347	348	349	
	350	351	352	353	354	
	355	356	357	358	359	
	360	361	362	363	364	
	365	366	367	368	369	
	370	371	372	373	374	
	375	376	377	378	379	
	380	381	382	383	384	
	385	386	387	388	389	
	390	391	392	393	394	
	395	396	397	398	399	
	400	401	402	403	404	
	405	406	407	408	409	
	410	411	412	413	414	
	415	416	417	418	419	
	420	421	422	423	424	
	425	426	427	428	429	
	430	431	432	433	434	
	435	436	437	438	439	
	440	441	442	443	444	
	445	446	447	448	449	
	450	451	452	453	454	
	455	456	457	458	459	
	460	461	462	463	464	
	465	466	467	468	469	
	470	471	472	473	474	
	475	476	477	478	479	
	480	481	482	483	484	
	485	486	487	488	489	
	490	491	492	493	494	
	495	496	497	498	499	
	500	501	502	503	504	
	505	506	507	508	509	
	510	511	512	513	514	
	515	516	517	518	519	
	520	521	522	523	524	
	525	526	527	528	529	
	530	531	532	533	534	
	535	536	537	538	539	
	540	541	542	543	544	
	545	546	547	548	549	
	550	551	552	553	554	
	555	556	557	558	559	
	560	561	562	563	564	
	565	566	567	568	569	
	570	571	572	573	574	
	575	576	577	578	579	
	580	581	582	583	584	
	585	586	587	588	589	
	590	591	592	593	594	
	595	596	597	598	599	
	600	601	602	603	604	
	605	606	607	608	609	
	610	611	612	613	614	
	615	616	617	618	619	
	620	621	622	623	624	
	625	626	627	628	629	
	630	631	632	633	634	
	635	636	637	638	639	
	640	641	642	643	644	
	645	646	647	648	649	
	650	651	652	653	654	
	655	656	657	658	659	
	660	661	662	663	664	
	665	666	667	668	669	
	670	671	672	673	674	
	675	676	677	678	679	
	680	681	682	683	684	
	685	686	687	688	689	
	690	691	692	693	694	
	695	696	697	698	699	
	700	701	702	703	704	
	705	706	707	708	709	
	710	711	712	713	714	
	715	716	717	718	719	
	720	721	722	723	724	
	725	726	727	728	729	
	730	731	732	733	734	
	735	736	737	738	739	
	740	741	742	743	744	
	745	746	747	748	749	
	750	751	752	753	754	
	755	756	757	758	759	
	760	761	762	763	764	
	765	766	767	768	769	
	770	771	772	773	774	
	775	776	777	778	779	
	780	781	782	783	784	
	785	786	787	788	789	
	790	791	792	793	794	
	795	796	797	798	799	
	800	801	802	803	804	
	805	806	807	808	809	
	810	811	812	813	814	
	815	816	817	818	819	
	820	821	822	823	824	
	825	826	827	828	829	
	830	831	832	833	834	
	835	836	837	838	839	
	840	841	842	843	844	
	845	846	847	848	849	
	850	851	852	853	854	
	855	856	857	858	859	
	860	861	862	863	864	
	865	866	867	868	869	
	870	871	872	873	874	
	875	876	877	878	879	
	880	881	882	883	884	
	885	886	887	888	889	
	890	891	892	893	894	
	895	896	897	898	899	
	900	901	902	903	904	
	905	906	907	908	909	
	910	911	912	913	914	
	915	916	917	918	919	
	920	921	922	923	924	
	925	926	927	928	929	
	930	931	932	933	934	
	935	936	937	938	939	
	940	941	942	943	944	
	945	946	947	948	949	
	950	951	952	953	954	
	955	956	957	958	959	
	960	961	962	963	964	
	965	966	967	968	969	
	970	971	972	973	974	
	975	976	977	978	979	
	980	981	982	983	984	
	985	986	987	988	989	
	990	991	992	993	994	
	995	996	997	998	999	

$$N_1 = 1, N_2 = 2, N_3 = 5$$

$$[N_1, N_2] \cap [N_1, N_3] = [N_1, N_3]$$

F.M. 21 ✓ F16 51x F21 51 ✓

$$N_a \text{ talesan } [N_a, N_b] = 1$$

Vamos a fallar un agrupamiento para la q se
algún $[N_i, N_j] = -1$



$P(k)$, es el numero de conexiones que forma

considere a los vecinos

$C(k)$ cuenta triangulos forma.

fila $(1, 2), (1, 3), (1, 4), (1, 5)$ Salida
1 0 -1 0

$(1, 2, 1), (3, 1), (4, 1), (5, 1)$ Entrada
0 1 0 0

Tome $k = 3$

$(2, 1), (2, 3), (2, 4), (2, 5)$
fila 0 0 0 0 1

Col 1 1 0 1

~~Script~~ script $(2, 5), (5, 2)$

✓ aqui (en columnas) se cuenta todos los conexiones
entre una sola fila $\therefore k = 3$

Considerar para el caso de que $N_i = N_j$

$$\frac{P(\text{no})}{P(\text{si})} = \frac{\ln i}{K_i(K_i - 1)}$$

$n = \text{union}$
 $n =$

Para que sea nula la ratio, un solo se analizará
 si se va a aplicar otra condición o no.

Para calcular $P(A)$, tomaremos de el Dataframe
 generado en la matriz inicial.

- El avance sera por fila y por columna
 Vamos a recorrerla en la fila uno, recordando
 se calcula así: contamos el numero de conexiones
 contando consigo mismo
- Ahora calcularemos el numero de la con-
 nexion de no triangulares.

Graficarán para cada uno.

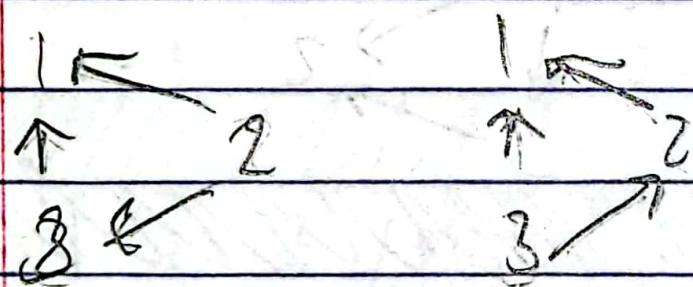
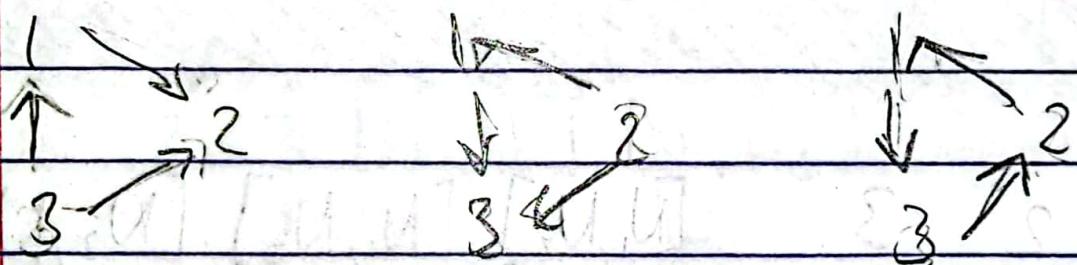
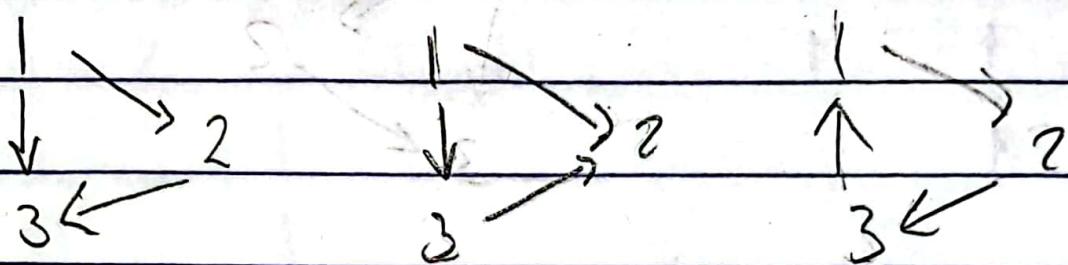
Porque total de conexiones que han
 hecho nodo

n no debe formar un dicho nodo

	k_{mat}	k_{in}	k	
1				
2				
:				
n				
<p>Ahora toca contar el total de k_i que hay</p>				
1	1 2 3 4	7	1 2 3 4	7
2			2	→
3			3	
4		4		↓
1	1 2 3 4	7	1 2 3 4	7
2			2	
3			3	
4		4		↓

Con un while podría ser que se cumpla, sin tener
que riguroso a hacer una cierta regresión

Ahora negriando el cálculo de triángulos (n)
con todas las combinaciones <Condicional n>



Va de la fila a la columna, poniendo flecha

			row, col	
1	2	3	$[N_1, N_2]$, $[N_1, N_3]$, $[N_2, N_3]$	ponee en que es necesario
1	2	3		

? Seguir los graficos, a como se ordenan en el script

1	2	3	1	2
1	1	1	1	2
2	1		3	
3				

A no, ya que el tiene si es necesario.

1	2	3	$[N_1, N_2]$, $[N_1, N_3]$, $[N_2, N_3]$	
1	1	1	1	2
2			1	2
3	1		3	



