

# Projecte d'Algorismia

**Integrants: Mireia Cano Pujol, Daniel Donate Durán, Daniel Losada Molina i Albert Ollé Sabaté**

## Experiments amb el Model Binomial

### *Com Executar*

Per executar els tests simplement s'ha de compilar l'arxiu `Graph_Erdos.cpp` i seguir les instruccions que apareixen en pantalla quan s'executa.

### *Metodologia i resultats*

Per estudiar el nombre esperat de components connexes en un graf generat aleatoriament a través del model binomial, hem escrit un petit programa que automatitza els tests. Aquest programa s'encarrega de, per a cada número de vèrtex  $N$ , recórrer totes les possibles probabilitats des de 0.01 fins a 0.99 i per a cada increment genera un graf i compta el número de components connexes 10 vegades i ho escriu en un arxiu csv on fem la mitja per obtenir un resultat representatiu.

Les  $N$  que hem escollit són 20, 60 i 100, els resultats es tallen a partir d'una probabilitat de 0.3 ja que en la continuació no hi havia cap variació i així s'entén millor el graf.

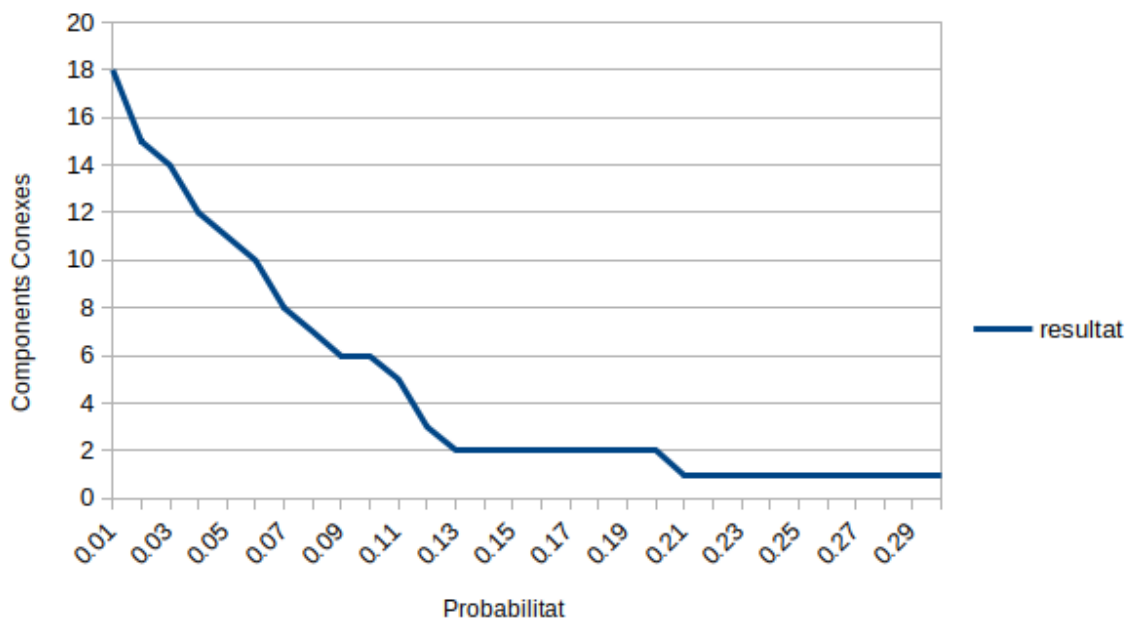


Fig. 1: Components Connexes esperades en un graf amb 20 vèrtex

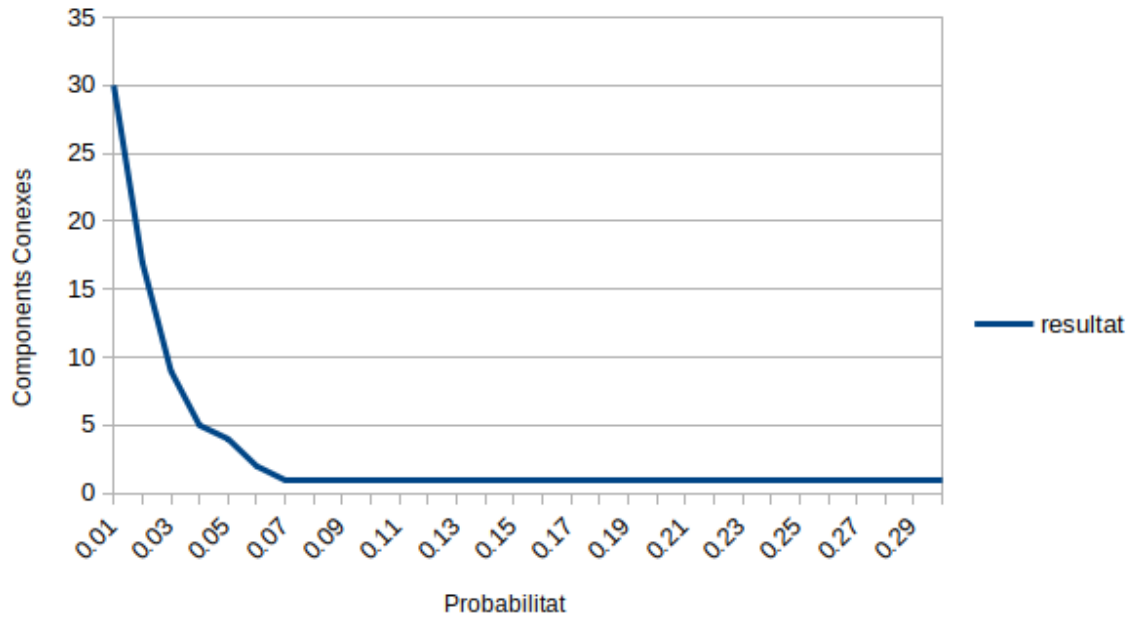


Fig. 2: Components Connexes esperades en un graf amb 60 vèrtex

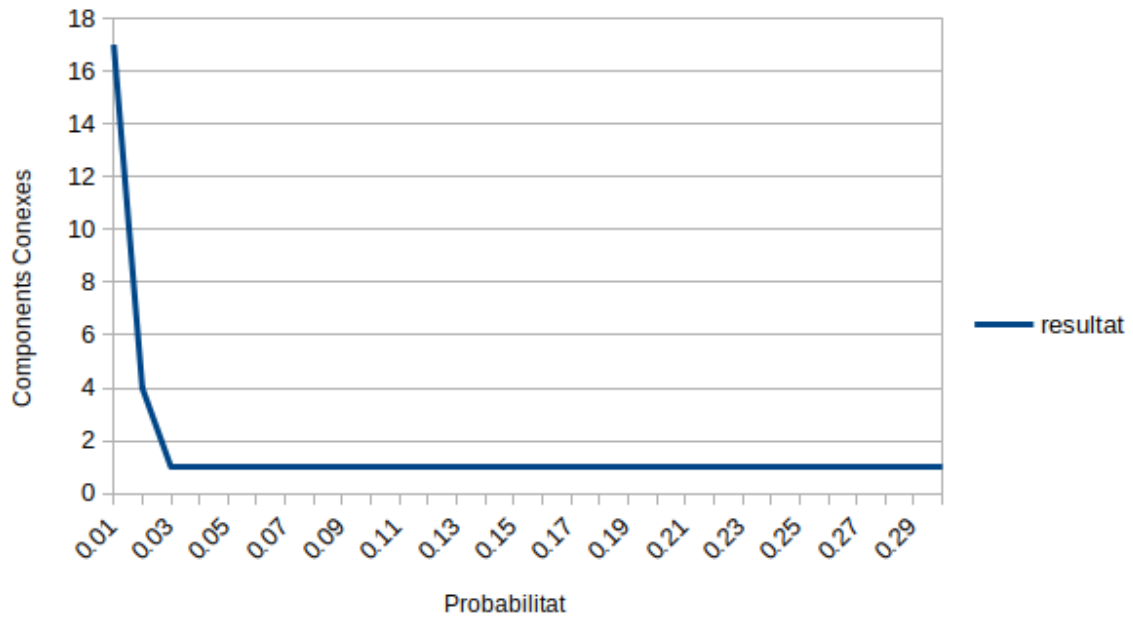


Fig. 3: Components Connexes esperades en un graf amb 100 vèrtex

# Experiments amb el model Geometric

## Com Executar

```
$ g++ -c randomGeometric.cc GraphRGG.cc
$ g++ -o graph.exe randomGeometric.o GraphRGG.o
$ ./graph.exe >NOM_FITXER.csv
```

En executar aquesta última comanda haurem d'introduir per consola el nombre de nodes del graf amb el que desitgem realitzar els experiments.

## Metodologia i resultats

Per estudiar el nombre esperat de components connexes en un graf RGG (Random geometric graph ( $G(n, r)$ )) hem utilitzat un programa que automatitza el test i escriu els resultats en un arxiu .csv. A aquest programa se li introdueix el número de vèrtex  $N$  i genera 100  $r$  aleatòries; per a cadascuna d'aquestes  $r$  es generen 10 grafs de la forma esmentada abans i es compten les components connexes que té. Les  $N$  que hem escollit són 20, 60 i 100. El nostre codi inicialment tenia limitada la generació de coordenades dels nodes a 100 (per tant el graf se situava en un espai de mida 100x100, on cada  $x$  i  $y$  tenia un rang de 0 a 100), i hem observat que, d'aquesta manera, per a molts valors de  $r$  els grafs resultants amb un gran nombre de nodes només tenien una component connexa. Així que hem repetit l'experiment augmentant el límit de coordenades a 200 per separar una mica els nodes.

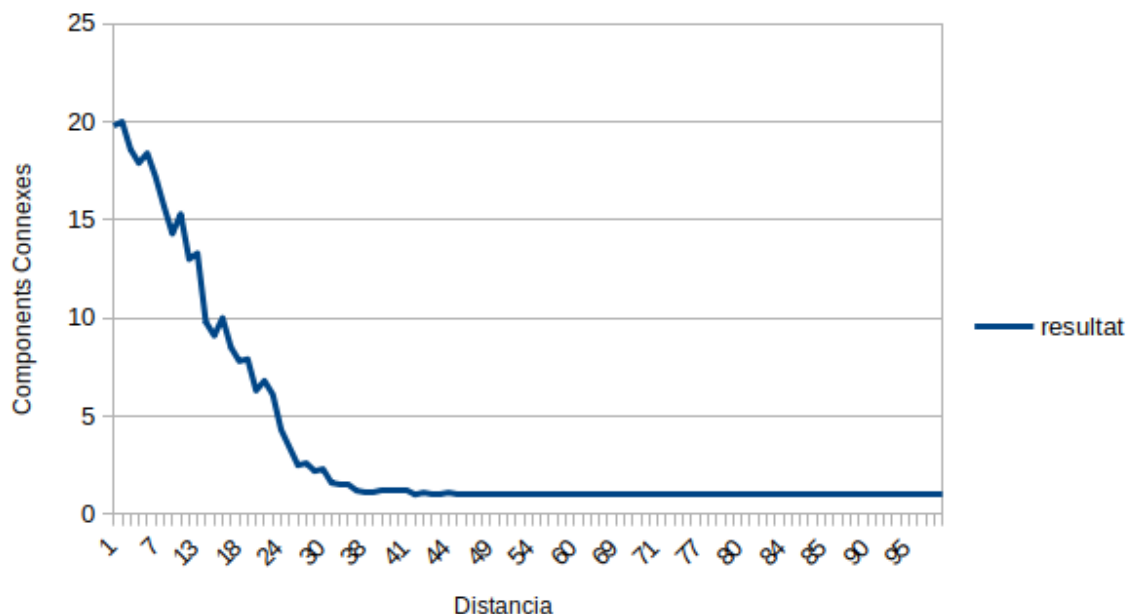


Fig. 4: Components Connexes esperades en un graf amb 20 vèrtex en un espai de 100x100

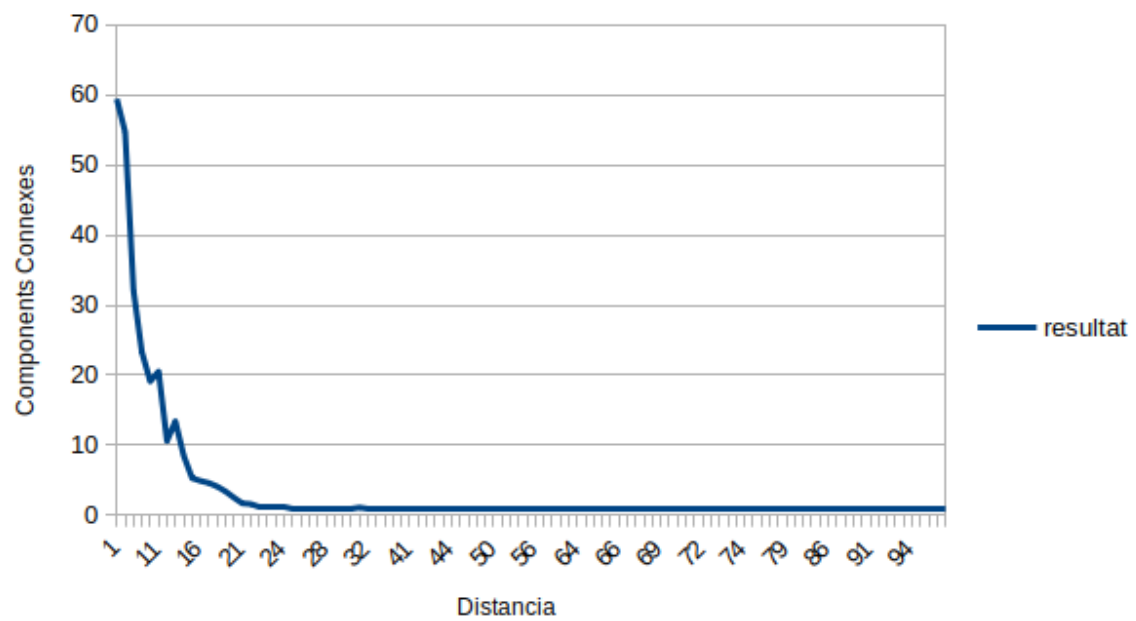


Fig. 5: Components Connexes esperades en un graf amb 60 vèrtex en un espai de 100x100

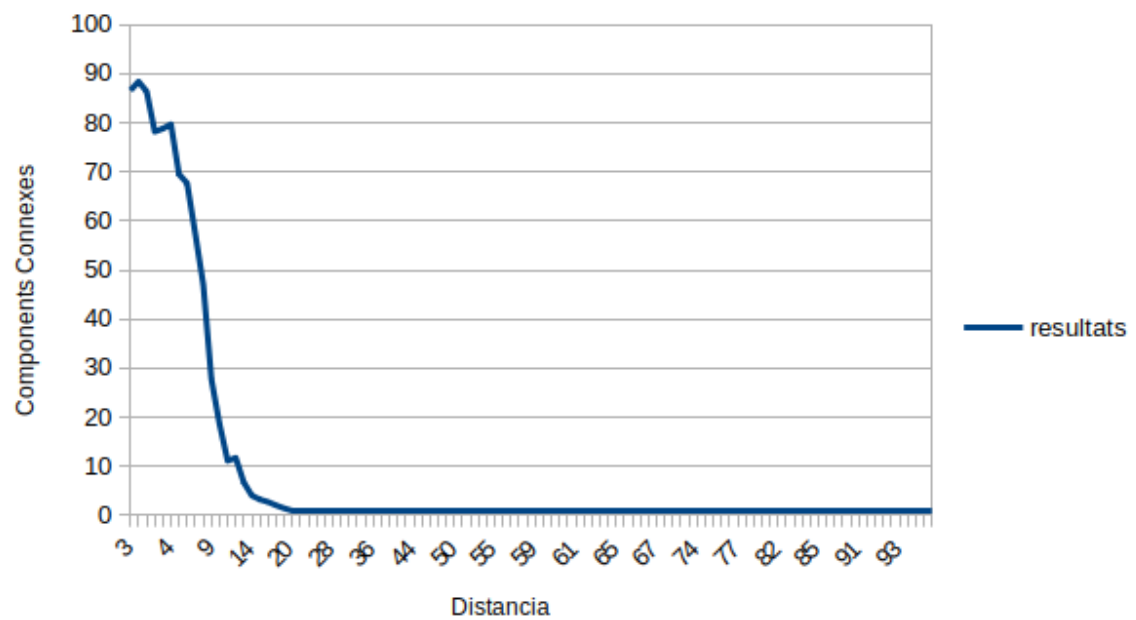


Fig. 6: Components Connexes esperades en un graf amb 100 vèrtex en un espai de 100x100

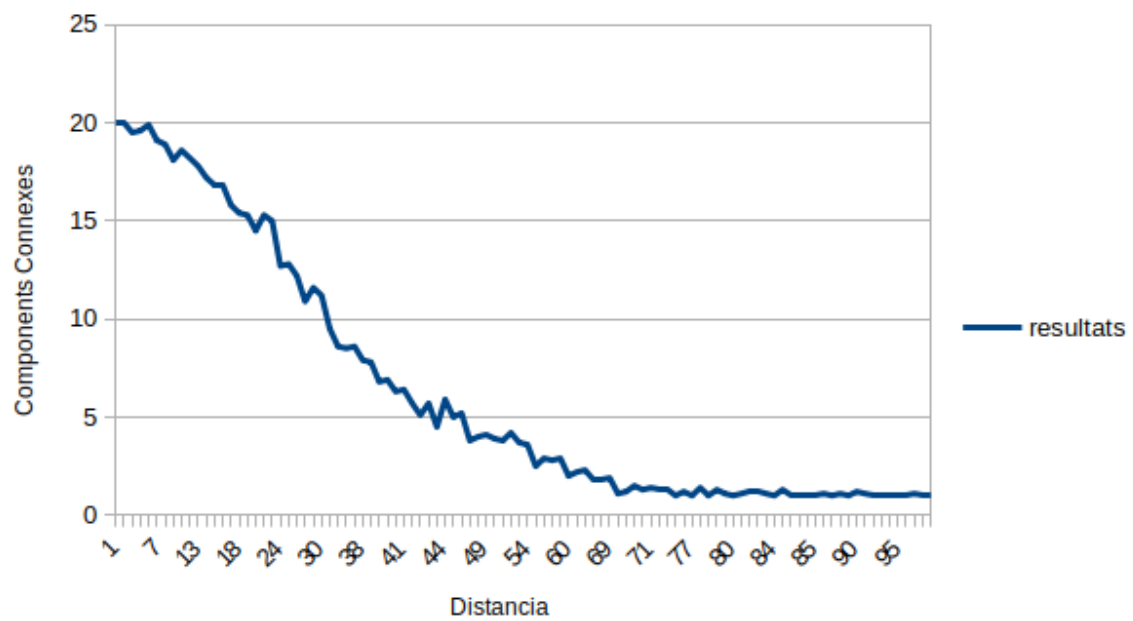


Fig. 7: Components Connexes esperades en un graf amb 20 vèrtex en un espai de 200x200

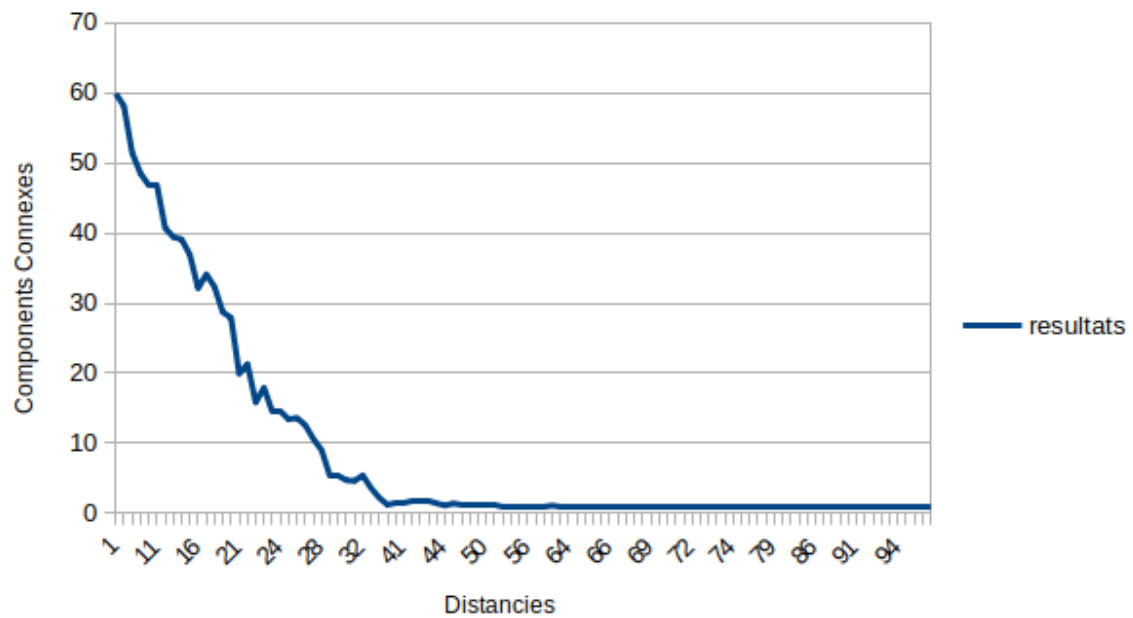


Fig. 8: Components Connexes esperades en un graf amb 60 vèrtex en un espai de 200x200

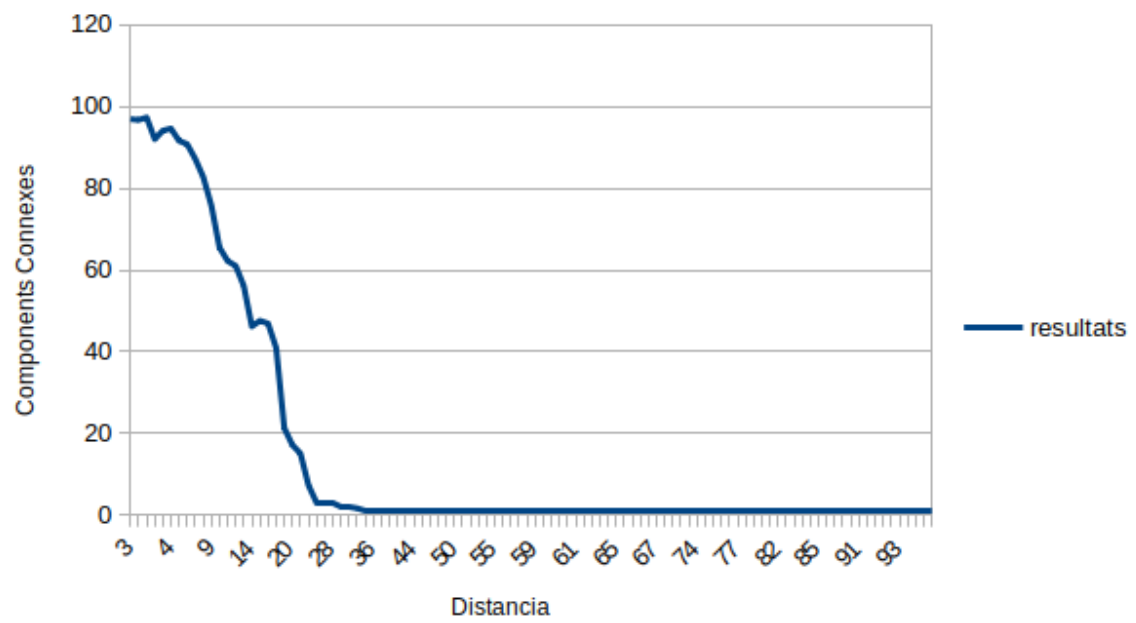


Fig. 9: Components Connexes esperades en un graf amb 100 vèrtex en un espai de 200x200