

Nom: Adrià Muro Gómez

NIU: 1665191

Nom: David Morillo Massagué

NIU: 1666540

Grafs: Pràctica 2

Tasca 1. Simulació cliques:

1. Com evoluciona la mida màxima dels grups que troba la vostra funció quan canvien els valors de n , m i s ? Penseu que, donat que estem en una simulació, per respondre aquesta pregunta haureu d'observar què passa en múltiples simulacions.

Per a fer aquesta prova s'han fet servir les primeres 5000 línies (arestes) del fitxer "email-Eu-core.txt", per a poder agilitzar el procés.

Incrementant els valors individualment, els resultats que ens donen són:

n : 10

m : 20

s : 5

Grups de 2 persones que s'han intercanviat almenys 10 missatges: 3577

Grups de 3 persones que s'han intercanviat almenys 10 missatges: 3853

Grups de 4 persones que s'han intercanviat almenys 10 missatges: 2395

Grups de 5 persones que s'han intercanviat almenys 10 missatges: 909

Grups de 6 persones que s'han intercanviat almenys 10 missatges: 232

Grups de 7 persones que s'han intercanviat almenys 10 missatges: 36

Grups de 8 persones que s'han intercanviat almenys 10 missatges: 2

Temps: 16.772828102111816 segons

n : 20

m : 20

s : 5

Grups de 2 persones que s'han intercanviat almenys 20 missatges: 3577

Grups de 3 persones que s'han intercanviat almenys 20 missatges: 554

Grups de 4 persones que s'han intercanviat almenys 20 missatges: 58

Grups de 5 persones que s'han intercanviat almenys 20 missatges: 5

Temps: 15.9883394241333 segons

n : 10

m : 50

s : 5

Grups de 2 persones que s'han intercanviat almenys 10 missatges: 3577

Grups de 3 persones que s'han intercanviat almenys 10 missatges: 4119

Grups de 4 persones que s'han intercanviat almenys 10 missatges: 2747

Grups de 5 persones que s'han intercanviat almenys 10 missatges: 1128

Grups de 6 persones que s'han intercanviat almenys 10 missatges: 308

Grups de 7 persones que s'han intercanviat almenys 10 missatges: 62

Grups de 8 persones que s'han intercanviat almenys 10 missatges: 9

Grups de 9 persones que s'han intercanviat almenys 10 missatges: 1

Temps: 16.35705256462097 segons

2. De nou, a mesura que creixen n i m , com creix el temps necessari per respondre a la primera pregunta?

Com veiem en l'apartat anterior, el temps necessari no varia en funció de n o m , encara que ho comprovem amb quantitats de milions d'unitats. Això és perquè la complexitat de l'algoritme no depèn fortament del nombre de n o m , sinó del nombre de vèrtexs en el graf inicial. Per això no veiem gairebé diferències en els temps de còmput.

Tasca 2. La lota $n < m$:

1. A mesura que creixen n i m , els resultats que dona el vostre programa s'ajusten al què hem vist a la part de combinatòria de l'assignatura? Perquè? Penseu que, com en la tasca anterior, estem davant d'una simulació...

Si que s'ajusten, a mesura que augmenta tant n com m el temps d'execució per trobar una solució es veu afectat, sobretot, l'augment del paràmetre de la longitud de la combinació a endevinar és el que produeix un major canvi respecte el temps.

2. De nou, a mesura que creixen n i m , com creix el temps necessari per respondre a la primera pregunta?

Per respondre aquesta pregunta, s'ha creat una taula amb els resultats, la combinació que s'ha utilitzat és $1^*(\text{longitud})$ i sempre amb els nombres del 0 al 9 disponibles

Longitud de la combinació	Segons fins que la combinació sigui guanyadora	Nombre de sortejos necessaris
1	0,0	12
2	0,0	123
3	0,0	867
4	0,0	3182
5	0,09332680702209473	19133
6	9,008795261383057	1578849
7	62,30197882652283	9153309
8	722,6323800086975	93002830

Tasca 3. És la nostra xarxa plana?

1. Expliqueu amb detall com funciona la vostra funció. Penseu que heu de poder explicar què fa cada una de les funcions que feu servir per obtenir la resposta.

```
def big_flat(G):

    n = len(G.nodes) # Nombre de vèrtexs

    if nx.is_planar(G): # Aquesta funció determina si el graf és pla o no

        return G # Si és pla retorna el graf sense canvis

    else: # Sinó

        while not nx.is_planar(G): # Mentre G no sigui un graf pla

            if n > 5: #Per K(3,3):

                for subgraf in combinations(G.nodes(),6): # Aquesta línia itera sobre tots els subgrafs de 6 vèrtexs
                    possibles en el graf G

                    SubG = G.subgraph(subgraf) # Transforma el subgraf en un subgraf del tipus networkx

                    if bipartite.is_bipartite(SubG): # Aquesta funció determina si el subgraf és bipartit

                        esquerra, dreta = bipartite.sets(SubG) # Separa els dos conjunts del graf bipartit

                        if len(esquerra)==3 and len(dreta)==3: # Determina si les longituds dels dos conjunts és 3. Si
                            es compleix, el subgraf és un K(3,3)

                            nodes = list(SubG.nodes) # Transforma els nodes del tipus networkx en una llista bàsica
                                de python

                            G.remove_node(nodes[0]) # Elimina un node del graf bipartit K(3,3) perquè el graf sigui
                                pla

                            if n > 4: #Per K(5)

                                for subgraf in combinations(G.nodes(),5): # Aquesta línia itera sobre tots els subgrafs de 5 vèrtexs
                                    possibles en el graf G

                                    subG= G.subgraph(subgraf) # Transforma el subgraf en un subgraf del tipus networkx

                                    if len(subG.edges())==10: # Mira si el subgraf de 5 vèrtexs té 10 arestes que significa que es tracta
                                        del graf complet K(5)

                                        nodes = list(subG.nodes) # Transforma els nodes del tipus networkx en una llista bàsica de
                                            python

                                        G.remove_node(nodes[0]) # Elimina un node del graf complet K(5) perquè el graf sigui pla

                                return G # Retorna el graf modificat perquè sigui pla
```

1. És la vostra funció capaç de donar la resposta correcta per al nostre graf de correus? Fins a quines mides de grafs (en termes de número de vèrtexs i arestes) arriba la vostra funció?

La nostra funció no és capaç de processar el graf de correus ja que la búsqueda de tots els subgrups per tal de trobar els possibles $K(5)$ o $K(3,3)$ utilitzant el nostre algoritme és molt lenta.

El nostre algoritme és capaç de processar un graf complet de 43 vèrtexs aproximadament, és a dir, 903 arestes aproximadament.