

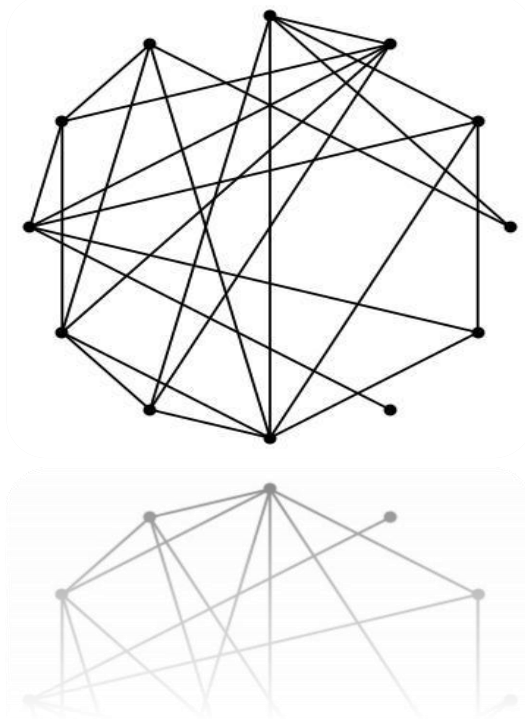


UNIVERSITÉ SIDI MOHAMED BEN ABDELLAH

FACULTÉ DES SCIENCES DHAR EL MAHRAZ
UNIVERSITÉ SIDI MOHAMED BEN ABDELLAH

2018/2019

TP 4: Les Graphes Aléatoires



Enseignant :

Pr. MOHAMED MEKNASSI

Réalisé par le groupe 6:

EL BAGHDADI MOHAMED

BERRAG AYOUB

SOMMAIRE

I. Introduction.....	2
II. Les Graphes Aleatoires.....	2
1. Graphe Nul ou Vide	2
2. Graphe Complet ou Plein	3
3. Densité d'un graphe	3
4. Ordre d'un graphe	3
5. Degré D'un Sommet	4
III. Génération d'un Graphe Aléatoire.....	4
1. Le matériel utilisé	7
2. Le logiciel utilisé	7
V. Annexe	7
VI. Questions De Recherche	9

I. Introduction

Dans ce Travail Pratiques, on va créer des graphes aléatoires sous le langage C++ en utilisant la fonction `rand ()` disponible sous la librairie `stdlib.h`. Un graphe se compose des arêtes et des sommets, on va générer les deux aléatoirement.

II. Les Graphes Aleatoires

En mathématiques, un graphe aléatoire est un graphe qui est généré par un processus aléatoire. Le premier modèle de graphes aléatoires a été popularisé par Paul Erdős et Alfréd Rényi dans une série d'articles publiés entre 1959 et 1968.

1. Graphe Nul ou Vide

En mathématiques, plus spécialement en théorie des graphes, un graphe nul désigne soit un graphe d'ordre zéro (i.e. sans sommets), soit un graphe avec sommets mais sans arêtes (on parle aussi dans ce dernier cas de graphe vide).

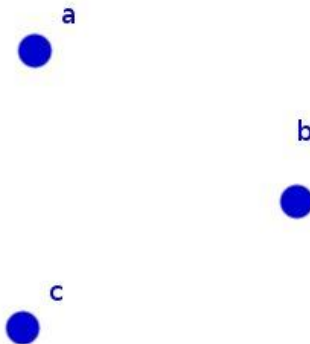


Figure 1:Exemple de graphe nul.

2. Graphe Complet ou Plein

En théorie des graphes, un graphe complet est un graphe simple dont tous les sommets sont adjacents, c'est-à-dire que tout couple de sommets disjoints est relié par une arête. Si le graphe est orienté, on dit qu'il est complet si chaque paire de sommets est reliée par exactement deux arcs (un dans chaque sens).

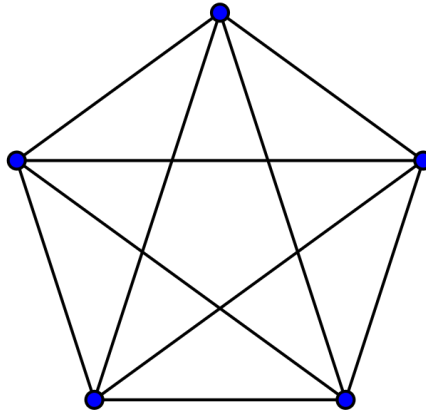


Figure 2: Graphe complet

3. Densité d'un graphe

En mathématiques, et plus particulièrement en théorie des graphes, on peut associer à tout graphe un entier appelé densité du graphe. Ce paramètre mesure si le graphe a beaucoup d'arêtes ou peu. Un graphe dense (dense graph) est un graphe dans lequel le nombre d'arêtes (ou d'arcs) est proche du nombre maximal, par exemple un nombre quadratique par rapport au nombre de sommets. Un graphe creux (sparse graph) a au contraire peu d'arêtes, par exemple un nombre linéaire. La distinction entre graphe creux et dense est plutôt vague et dépend du contexte.

4. Ordre d'un graphe

Nombre de sommets d'un graphe. Le graphe ci-dessous est un graphe d'ordre 6.

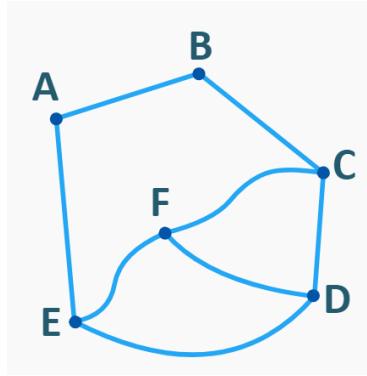


Figure 3: Graphe d'ordre 6

5. Degré D'un Sommet

En mathématiques, et plus particulièrement en théorie des graphes, le degré (ou valence) d'un sommet d'un graphe est le nombre de liens (arêtes ou arcs) reliant ce sommet, avec les boucles comptées deux fois.

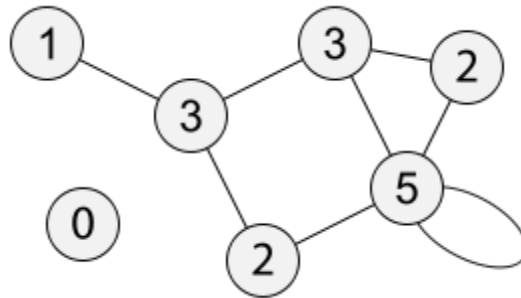


Figure 4: degré des sommets

Un graphe G non orienté où on a indiqué le degré de chaque sommet sur ce sommet. Dans ce graphe, le degré maximal est 5 et le degré minimal est 0.

III. Génération d'un Graphe Aléatoire

On écrit un programme en C++ qui permet de générer un graphe aléatoire, ce programme contient les fonctions suivantes :

`genererUnGraphe(int NOE, int NOV)` qui prend deux arguments pour spécifier le nombre des sommets et des arêtes et génère par suite le graphe. Ce graphe permet aussi d'afficher le graphe.

La fonction main qui permet de générer deux nombres aléatoires en utilisant la fonction rand () de la librairie stdlib, ces nombres (nombre des sommets et des arêtes) vont être utilisé comme des arguments de la fonction genererUnGraphe().

Pour chaque sommets et arête générer on associe un poids aléatoirement. On change le nombre des sommets à chaque fois pour obtenir des différents graphes.

On compile et on exécute le programme, des exemples de résultat sont ci-dessous :

```

generation d un graphe aleatoire:
Le graphe contient 12 sommets.
Le graphe contient 53 aretes.
Le graphe genere aleatoirement est:
1-> < 4 11 9 7 10 6 12 5 8 3 >
2-> < 4 12 11 3 7 9 >
3-> < 8 10 12 7 5 11 2 6 1 >
4-> < 2 1 12 11 10 9 8 5 >
5-> < 11 6 10 12 3 9 1 4 >
6-> < 5 11 10 7 1 8 9 3 12 >
7-> < 12 8 3 10 1 6 11 2 >
8-> < 3 12 7 10 9 6 1 4 >
9-> < 11 10 12 1 5 4 8 2 6 >
10-> < 3 9 5 8 11 12 7 6 1 4 >
11-> < 5 9 10 2 1 3 6 7 4 12 >
12-> < 2 3 7 8 4 9 10 5 1 11 6 >
-----
Process exited after 7.019 seconds with return value 0
Press any key to continue . . .

```

Figure 5: exemple 1

Exemple de graphe plein (80% des arêtes).

N le nombre des sommets est égale a 12, un graphe complet possède $N(N-1)/2$ arêtes.

Ce graphe possède 53 arêtes, donc il contient $(53*100)/66 = 80 \%$.

```

generation d un graphe aleatoire:
Le graphe contient 50 sommets.
Le graphe contient 41 aretes.
Le graphe genere aleatoirement est:
 1-> < 20 7 >
 2-> < 44 >
 3-> < 5 >
 4-> < 43 45 12 >
 5-> < 3 >
 6-> < 15 41 >
 7-> < 41 1 >
 8-> < 38 >
 9-> < 13 >
10-> < 24 >
11-> < sommet isloe! >
12-> < 28 23 4 >
13-> < 9 18 45 >
14-> < sommet isloe! >
15-> < 6 42 43 >
16-> < sommet isloe! >
17-> < 19 29 >
18-> < 35 13 >
19-> < 17 48 >
20-> < 1 39 >
21-> < 30 >
22-> < 33 27 >
23-> < 12 >
24-> < 34 10 35 >
25-> < 29 >
26-> < sommet isloe! >
27-> < 22 >
28-> < 12 43 >
29-> < 25 17 >
30-> < 42 21 49 >
31-> < sommet isloe! >
32-> < 46 >
33-> < 22 >
34-> < 24 >
35-> < 18 24 >
36-> < 50 41 >
37-> < 42 >
38-> < 8 >
39-> < 20 43 >
40-> < sommet isloe! >
41-> < 36 7 6 >
42-> < 46 37 15 30 >
43-> < 28 4 39 15 >
44-> < 2 >
45-> < 4 13 >
46-> < 32 42 48 >
47-> < 49 >
48-> < 46 19 >
49-> < 47 30 >
50-> < 36 >
-----
Process exited after 6.493 seconds with return value 0
Press any key to continue . . .

```

Figure 6:exemple 2

Exemple de graphe vide (3,3%).

IV. Environnement du travail

1. Le matériel utilisé

Processor:	Intel(R) Core(TM) i5 CPU	M 540 @ 2.53GHz	2.53 GHz
Installed memory (RAM):	4.00 GB	(3.80 GB usable)	
System type:	64-bit Operating System		

Figure 7: le matériel utilisé

2. Le logiciel utilisé

Nous avons utilisé le logiciel DEV C++ qui contient un compilateur de C++.

V. Annexe

```
#include<iostream>
#include<stdlib.h>

using namespace std;

// Une fonction qui genere un graphe aleatoire.
void genererUnGraphe(int NOE, int NOV)
{
    int i, j, edge[NOE][2], count;
    i = 0;
    // construire les aretes entre les sommets.
    while(i < NOE)
    {
        edge[i][0] = rand()%NOV+1;
        edge[i][1] = rand()%NOV+1;

        if(edge[i][0] == edge[i][1])
            continue;
        else
        {
            for(j = 0; j < i; j++)
```



```

        {
            if((edge[i][0] == edge[j][0] && edge[i][1] == edge[j][1]) || (edge[i][0]
== edge[j][1] && edge[i][1] == edge[j][0]))
                i--;
        }
    }
    i++;
}

// affichage du graphe.
cout<<"\n Le graphe genere aleatoirement est: ";
for(i = 0; i < NOV; i++)
{
    count = 0;
    cout<<"\n\t"<<i+1<<"-> { ";
    for(j = 0; j < NOE; j++)
    {
        if(edge[j][0] == i+1)
        {
            cout<<edge[j][1]<<" ";
            count++;
        }
        else if(edge[j][1] == i+1)
        {
            cout<<edge[j][0]<<" ";
            count++;
        }
        else if(j == NOE-1 && count == 0)
            cout<<"sommet isole!";
    }
    cout<<" }";
}
}

int main()
{
    int n, i, e, v;

    cout<<"generation d un graphe aleatoire: ";

    // affecter des sommets et des aretes aleatoires.
    v = 11+rand()%10;
    cout<<"\n Le graphe contient "<<v<<" sommets.";
    e = rand()%((v*(v-1))/2);
    cout<<"\n Le graphe contient "<<e<<" aretes.";

    // une fonction qui genere un graphe aleatoire avec e aretes et v sommets.
    genererUnGraphe(e, v);
}

```

}

VI. Questions De Recherche

- Quel sont les courbes qui existent pour les complexités ?

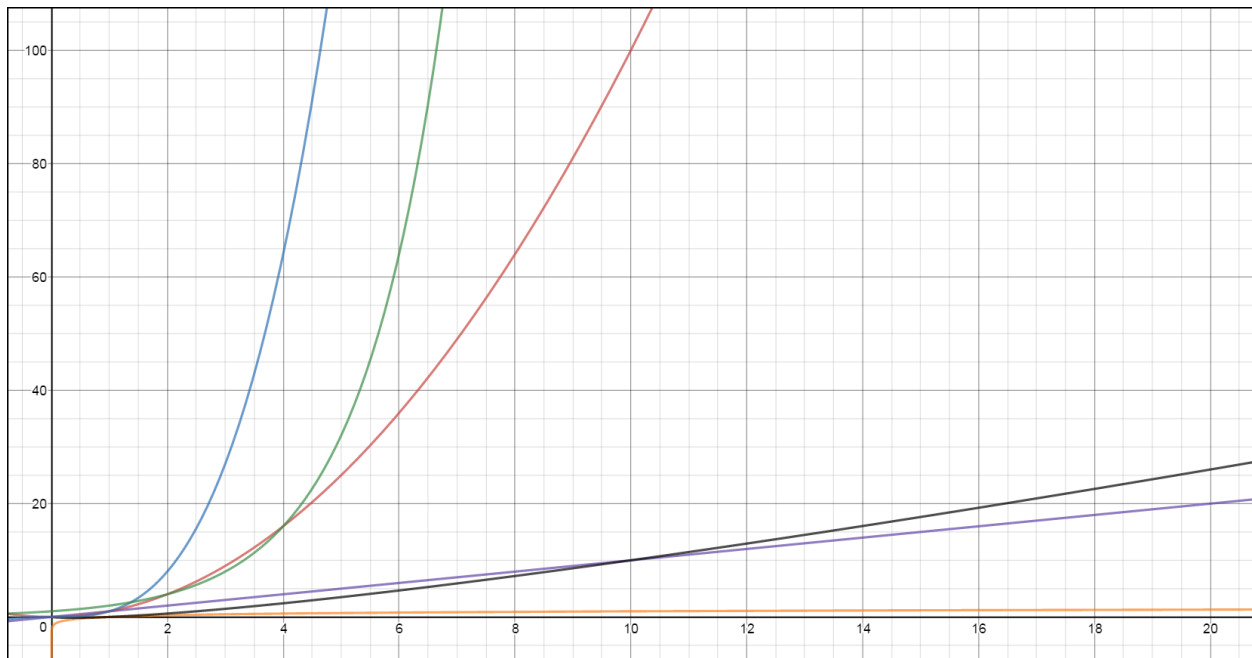


Figure 8: Représentation graphique des fonctions de référence

Du "bas vers le haut" on a les fonctions $\log(x)$ $x \log(x)$, x^2 , x^3 , $2x$.

- **Donner la définition de l'acronyme VLSI.**

L'intégration à très grande échelle (ou VLSI pour Very-Large-Scale Integration en anglais) est une technologie de circuit intégré (CI) dont la densité d'intégration permet de supporter plus de 100 000 composants électroniques sur une même puce.

- **Donner la définition de l'acronyme VHDL.**

VHDL est un langage de description de matériel destiné à représenter le comportement ainsi que l'architecture d'un système électronique numérique. Son nom complet est VHSIC1 Hardware Description Language.