图与网络：Lab 2 报告

小组成员：
黎锦灏 518021910771
沈刘 518030910114
程欣雯 518021910031

GitHub repo 链接：
https://github.com/ljh2000/MATH6010_Graph-and-Network

一、问题描述

- 定理：
  对一个有 n 条边的完全图进行二染色，最多有$\binom{n}{4}2^{-5}$ 单色 K_4。
- 实现内容：
  用贪心算法实现边染色，每次选取期望同色 K_4 数目更小的方案，使得最终得到的同色 K_4 的数目小于等于期望值。

二、算法流程

Step 1:
生成包含 n 个节点的完全图，记录每条边以及其两个端点；

Step 2:
遍历 **Step 1** 中记录的每一条边，在图中找到所有包含这条边的所有 K_4；

Step 3:
分别计算这条边染黑色和染白色对产生单色 K_4 的影响，计算 I_{K_4} 指示变量，并求和得到 W_{black} 和 W_{white}；

Step 4:
根据贪心策略，由指示变量 W_{black} 和 W_{white} 的值决定这条边的染色结果：

- 若 $W_{black} < W_{white}$，则将当前边染为黑色；
- 若 $W_{black} > W_{white}$，则将当前边染为白色。

三、程序框架

程序分为以下几个部分：

1. 生成图网络函数 `init_edge()`：
   这个函数根据条件生成由 n 个节点组成的完全图，为保证实验结果的可靠性，每次实验先对 $\binom{n}{2}$ 条边做随机排列，以排除染色顺序对实验的影响。

```
void init_edge(){
    srand(time(NULL));

    int ecnt = 0;
    for(int i = 1; i <= n; i++)
        for(int j = i+1; j <= n; j++) {
            ecnt ++;
            e[ecnt].x = i;
            e[ecnt].y = j;
        }

    for(int i = 1; i <= m; i++) id[i] = i;
    random_shuffle(id+1,id+m+1);
}
```

2. I_{K_4}计算函数`calc_Ik4()`：
   这个函数输入六条边的信息（黑色边、白色边、未染色边），计算得到可以得到同色 K_4 的期望：

- 对于已经同时存在黑色边和白色边的图，同色 K_4 的期望为 0；
- 对于还未染色的K_4，同色 K_4 的期望为 2^{-5}；
- 对于已经部分染色的K_4，同色 K_4 的期望为 2^{r-6}，其中 r 表示已染色边的数目。

```
// black:1  white:2  null:0
double calc_Ik4(int col1, int col2, int col3, int col4, int col5, int col6){
    int k4_col = col1 | col2 | col3 | col4 | col5 | col6;
    int r = (col1>0) + (col2>0) + (col3>0) + (col4>0) + (col5>0) + (col6>0);

    // black and white : 0
    if(k4_col == 3) return 0;

    // null : 2^(-5)
    if(r == 0) return pow2[5];

    // black or white : 2^(r-6)
    return pow2[6-r];
}
```

3. 染色函数`color_edge()`：
   这个函数遍历所有 K_4，每次分别重新计算与当前边有关的 K_4 成为同色 K_4 的期望 W_{black} 和 W_{white}，每次选取期望小的方案进行染色，本算法的复杂度为 $O(n^4)$。

```
void color_edge(){
    ans = avg_ans;
    for(int idx = 1; idx <= m; idx++) {
        int x = e[ id[idx] ].x;
        int y = e[ id[idx] ].y;

        // calculate W_black and W_white
        double W_black = ans , W_white = ans;

        for(int i = 1; i <= n; i++) {
            if(x == i || y == i) continue;
            for(int j = i+1 ; j <= n;j++) {
                if(x == j || y == j) continue;

                // K4: x y i j
                double current_Ik4 = calc_Ik4(col[x][y], col[x][i], col[x][j], col[y][i], col[y][j], col[i][j]);

                W_black -= current_Ik4;
                W_black += calc_Ik4(1, col[x][i], col[x][j], col[y][i], col[y][j], col[i][j]);

                W_white -= current_Ik4;
                W_white += calc_Ik4(2, col[x][i], col[x][j], col[y][i], col[y][j], col[i][j]);

            }
        }

        if(W_black <= W_white) {
            ans = W_black;
            col[x][y] = col[y][x] = 1;
        }
        else {
            ans = W_white;
            col[x][y] = col[y][x] = 2;
        }
    }
}
```

4. 主函数`main()`

   主函数可以根据用户输入来确定生成的完全图的大小。

```
int main(int argc,char *argv[])
{
    //n = 200;
    //n = 6;
    if (argc == 1){
        n = 200;
    } else {
        n = atoi(argv[1]);
    }
    cout << "current edge num : "<< n << endl ;
    m = n*(n-1)/2;

    // preprocess powers of two
    pow2[0] = 1;
    for(int i = 1;i <= 6; i++) pow2[i] = pow2[i-1] / 2.0;
    //for(int i = 1;i <= 6; i++) cout << pow2[i] << " " ;

    init_edge();

    avg_ans = n * (n-1) * (n-2) * (n-3) / 24;
    avg_ans *= pow2[5];

    printf("Expected number of K4 in the same color : %10.0lf\n",avg_ans);

    color_edge();

    printf("Final answer : %10.0lf\n",ans);
    cout<<endl;

    return 0;
}
```

四、结果分析

为主函数编写测试脚本，测试脚本输入从 1 到 200 个节点的完全图。

```
# compile
g++ Edge_Coloring.cpp -o Edge_Coloring

# test
for num in {1..200}
do
./Edge_Coloring $num
done
```

测试结果如下图：
可以看到，使用条件概率的方法计算得到的同色 K_4小于理论最大值，实验验证定理成立。

```
current edge num : 192
Expected number of K4 in the same color :    1714702
Final answer :    1595506

current edge num : 193
Expected number of K4 in the same color :    1750992
Final answer :    1629602

current edge num : 194
Expected number of K4 in the same color :    1787856
Final answer :    1664248

current edge num : 195
Expected number of K4 in the same color :    1825298
Final answer :    1700362

current edge num : 196
Expected number of K4 in the same color :    1863325
Final answer :    1736361

current edge num : 197
Expected number of K4 in the same color :    1901943
Final answer :    1772540

current edge num : 198
Expected number of K4 in the same color :    1941158
Final answer :    1810342

current edge num : 199
Expected number of K4 in the same color :    1980977
Final answer :    1847905

current edge num : 200
Expected number of K4 in the same color :    2021405
Final answer :    1886300
```