算法分析与设计

**实验六 普里姆算法与克鲁斯卡尔算法**

学号：11603080406 姓名：李鑫瑜

## 一、Prim

### 1. 数据规模与时间统计表

|  |  |
| --- | --- |
| **n** | **时间（s）** |
| **1000** | **0.304** |
| **2000** | 1.409 |
| **3000** | 1.567 |
| **4000** | 4.401 |
| **5000** | 3.977 |
| **6000** | 5.507 |
| **7000** | 8.338 |
| **8000** | 11.203 |
| **9000** | 11.757 |
| **10000** | 20.057 |

## 2. 运行时间与规模图

## 3. 算法分析

Prim算法的时间复杂度（邻接矩阵）为O（(E log(V)）（E为边数，V为顶点），空间复杂度为O（V^2），在数据量达到十万时内存就溢出了，因为他要用的n\*n的表格实在是太大了。

普里姆算法的时间消耗还是很可观的，在数据量在1W以内时间不超过25秒。

## 二、Kruskal

### 1. 数据规模与时间统计表

|  |  |
| --- | --- |
| **n** | **时间（s）** |
| **10** | **0.016** |
| **100** | 0.024 |
| **1000** | 0.216 |
| **10000** | 0.216 |
| **100000** | 0.472 |
| **1000000** | 7.047 |
| **10000000** | 17.030 |
| **100000000** | 16.265 |

## 2. 运行时间与规模图

## 3. 算法分析

克鲁斯卡尔算法的时间复杂度也为O（Elog(V)），时间效率非常不错，数据量在10000时才消耗不到2S，而且空间消耗也非常少。关于普里姆与克鲁斯卡尔，一个对于稀疏图效率更高，一个对于稠密图，但对于数据量大的数据还是推荐克鲁斯卡尔。

## 三、带备忘录的递归算法

### 1. 数据规模与时间统计表

|  |  |
| --- | --- |
| **n** | **时间（s）** |
| **1000** | 0.220 |
| **2000** | 0.015 |
| **3000** | 1.951 |
| **4000** | 2.654 |
| **5000** | 7.094 |
| **6000** | 6.826 |
| **7000** | 7.381 |
| **8000** | 10.807 |
| **9000** | 12.547 |
| **10000** | 栈溢出 |

## 2. 运行时间与规模图

## 3. 算法分析

伪递归（备忘录）算法因为记录了每层递归的值，所以对于每个值不需要层层调用来算，因此比递归要省时一点，它的最坏时间复杂度为O（2的n次方），最好为O（n）,所以也并不稳定，耗时也比动态规划长。