

版本三基本实现

概述

1. 图结构和算法实现和版本二基本没有区别，但是变得更加模块化，其他的图算法如强连通分量、Page Rank等均可继承后，重载一些虚函数来实现整体算法。
2. 新增了数据的读取、图结构展示、结果展示的函数。

实现

1. Template.cpp

1. 图结构，如版本二，不再赘述。
2. 类中的数据

```
vector<struct Vertex> vertices;    // graph
int process[MAXN][2];             // record algorithm process
int update[MAXN][2];              // update info: distance + parentNum
int iter;                         // calculate algorithm complexity
int processNum;
```

3. 类中的函数

展示图结构：

```
1. void Display()
```

模块化图算法：

```
2. void Algorithm()
3. void Algorithm_M(int missGroup, int loss)
```

一些虚函数，用于继承类后实现特定的图算法：

```
4. virtual void Init(string filename){}
5. virtual void Relax(int index){}
6. virtual bool UpdateInfo(vector<struct Vertex> &vertices, int
update[][2]){}
7. virtual void ShowResult(){}
```

2. Template_distance.cpp

只需要继承后重载以上四个虚函数即可，注意数据输入的形式，每一行是三个整数：起始点序号、边权重、终止点序号，之后在 main 函数中直接利用即可。

```
int main(int argc, char const *argv[])
{
    ShortestPath test1;
    test1.Init("input.txt");
    // test1.Display();
    // test1.Algorithm();
    test1.Algorithm_M(2, 2);
    test1.ShowResult();

    return 0;
}
```

运行

1. g++ template2.cpp template_distance2.cpp -o outname
2. input2.txt 是所给的数据样例，其中第一行是节点个数和边的个数，若使用其他数据，可在main函数中修改。