2016-7-1

关于**优先队列**(其实没什么可写的233)

priority\_queue< int, vector<int>, less<int> > q;

less 可用 greater 替代

如果要自定义则需要

struct cmp{

bool operater()(myType a, myType b){

retrun a < b;

}

}

priority\_queue< int, vector<int>, cmp > q;

2016-7-2

可以用map来记录点状态，适用于无限大地图但只有有限个点可以使用的情况（没办法开状态数组）

map<pair<int,int>,int>m;

map中用自定义类型时，需要重载 <

struct node{

int x, y, z;

node(int a, int b, int c):x(a),y(b),z(c){}

bool operator < (const node & node1) const{

if (x == node1.x && y == node1.y)

return z < node1.z;

else if (x == node1.x)

return y < node1.y;

return x < node1.x;

}

};

2016-7-11

复习了一下图论的简单知识

最短路Dijkstra算法使用优先队列实现

SPFA算法，用来寻找负权环，使用队列保存每次更新的点，若某个点更新次数超过n次，则出现负权

2016-7-12

学习了字典树，具体在第一场个人赛中有代码。

2016-7-13

Longlong在做位移运算时需要声明类型 : numLL << k;

Next\_permutation() 可以快速的求出全排列

2016-7-15

未优化的next数组可以求解循环节，对于一个字符串， m - next[m] 是他的最小循环节，同时, m-next[next[m]] 等也是循环节