# 最短路及其优化

## Dijkstra

O(N^2)

->Heap-Dijkstra: O(M\*logN)

## Bellman-ford

O(N\*M)

不忍直视的时间复杂度

## SPFA

O(k\*M)

队列优化的Bellman-ford，k的期望值为2;

百度告诉我们，特殊情况下会退化到Bellman-ford...

时间复杂度仍旧不忍直视;

->Stack-SPFA

负环图派上用场;

有的没节操的卡队列的题卡不住栈;

->SLF-SPFA

优化约15%~20%

如果当前插入元素小于队首元素，插入到队首，否则插入到队尾;

->LLL-SPFA

优化约50%;

不常用;

->SLF+LLL-SPFA

->Heap-SPFA

Priority-queue + pair实现

实际上在正权图中就是Heap-Dijkstra

O(M\*logN)

注意:

存在负权边的图中特殊数据下的退化!

## 代码实现

int spfa(int s)

{

int re=0;

memset(d,0x3f,sizeof d);

memset(v,0,sizeof v);

q.push\_back(s);d[s]=0;v[s]=1;sum=0;size=1;

while(!q.empty())

{

int x,y;

while((x=q.front()))

{

if(d[x]\*size<=sum)

break;

q.pop\_front();

q.push\_back(x);

}

q.pop\_front();

re++;

sum-=d[x];

size--;

v[x]=0;

for(int i=head[x];i;i=next[i])

{

if(d[y=to[i]]>d[x]+val[i])

{

d[y]=d[x]+val[i];

if(v[y]==0)

{

if(!q.empty()&&d[y]<=d[q.front()])

q.push\_front(y);

else

q.push\_back(y);

v[y]=1;size++;sum+=d[y];

}

}

}

}

return re;

}

int spfa(int s)

{

int re=0;

memset(d,0x3f,sizeof d);

q.push(make\_pair(0,s));d[s]=0;

while(!q.empty())

{

int x=q.top().second,y=-q.top().first;

q.pop();

if(y!=d[x]) continue;

re++;

y=0;

for(int i=head[x];i;i=next[i])

{

if(d[y=to[i]]>d[x]+val[i])

{

d[y]=d[x]+val[i];

q.push(make\_pair(-d[y],y));

}

}

}

return re;

}

## 练习题

<http://poj.org/problem?id=2159>

差分约束

如果一个系统由n个变量和m个约束条件组成，其中每个约束条件形如

xj-xi<=bk(i,j∈[1,n],k∈[1,m]),则称其为差分约束系统(system of difference constraints)。亦即，差分约束系统是求解关于一组变量的特殊不等式组的方法。

求解差分约束系统，可以转化成图论的[单源最短路径](http://baike.baidu.com/view/1939816.htm" \t "_blank)（或最长路径）问题。

——百度百科

数据范围:

*N* and *M* not exceeding 30 000 and 150 000 respectively.

测试结果:

Heap+SPFA: 610ms

Stack+SPFA: 516ms

Else TLE.

<http://poj.org/problem?id=4046>

题干及测试数据请自行Baidu

题意:

城市中有许多餐厅。已知它们分别的消费价格以及一些餐厅之间的打车费用。我们的主人公要和朋友们从学校打车回家，为了不丢脸，他一定要在途中最贵的那家餐厅请客吃饭，但照顾到钱包的问题，他希望所有消费尽可能小。请你帮他解决他至少要带多少钱。

多组测试数据。每组测试数据中给出多组询问，对于每一组询问，求最小花费。

解法:

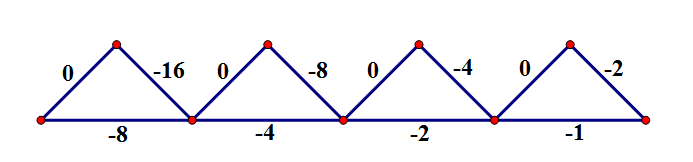
枚举第i个餐厅，则对于询问u,v，答案为min(dist[i,u]+dist[i,v]+cost[i]);

数据范围:

餐厅数n<=1000，边数m<=20000，询问数q<=20000.

每一份花销cost<=2\*10^9.

## 没节操的特殊数据



<三观尽毁.zip>

http://acm.hdu.edu.cn/showproblem.php?pid=4889

裸SPFA: 松弛n次

LLL+SPFA: 松弛4\*x次

SLF+SPFA/Stack-SPFA/Heap+SPFA: 松弛2^(x+1)+2^(x)-2次

X为三角形个数