最短路总结

最近刷了点最短路，就写一个总结吧，太系统理论的暂时还没有能力讲出来，不倒是可以总结几种最短路题目的方法。

那个三种虽短路的算法就不复习了，floyd，dijkstra，spfa 应该都会。

（一）普通最短路：

这一类题目一般来说图的性质很明显，不是有向图，就是无向图，而且题意也会明确的告诉你求从某一点到某一点的最短距离，通常都是模板题，很好写，就不多说了，网上也有很多。

（二）基于最短路的一些普通变形：

这些题目的很明显图，也很清晰，但是往往要求的并不是最短距离，而是在最短路的过程中更改其中某些属性，来获得答案，需要能够联想到最短路的性质还有最短路求解的过程中的一些变化，各种题目较为灵活。比如 POJ 2253 Frogger。这道题题意就不细说了，就是求最小的最大跳跃距离。这个为什么可以联想到最短路，因为最短路在求解释就是找的当前最短路的路径来更新即将到达的点的最短路径，那么也就可以找当前最小的最长路径来更行新的路径，Floyd 即可。关键代码如下：

```
for (int k=1;k<=n;k++){
    for (int i=1;i<=n;i++){
        for (int j=1;j<=n;j++){
            d[i][j]=min(d[i][j],max(d[i][k],d[k][j]));
        }
    }
}
```

与之相似的还有求路径上的最大容量 POJ 1797 Heavy transportation（不是网络流）等一系列的最短路上的某一个具有某种特征的值，需要增加一些练习，这种题目一般 Floyd 做。

（三）找环问题。

最短路之中一般是不想看到环的存在的，但是在某些题目中，则需要找环，因为环可以起到一个无限循环的作用，例如 POJ 1860 Currency Exchange，这个就是希望你能够找到一个环，使得能够获得利益（能够获利即可）。为什么环可以达到这个目的，我们来假设：如果经过一条边，可以让自身的当前利益增加和减少，那么从起点出发，再回到起点是肯定希望自身的利益是增加的，那么你并不能知道如果直接走一个来回是否能够达到这个目的。但是有环就不同了。假设点 A 是环的一个点，当从非环内点走到点 A 时，自身的利益肯定会有一个改变值 X，然后再从点 A 返回起点也会有一个改变值 Y，（X,Y 有正负）我们希望 X+Y>0，这样利益就会增加，但是这两个值是很难确定的，所以我们换个想法，不是有环么（假设是正环），那我们就一边一边的走环，让我们的利益增加的尽量多，那么反正回去的路上利益的降低是一个有限的值，但是增加的是无限得，所以找到环，那么问题就有解了。

至于怎么找环，当然是用 spfa。只要增加一个计数的数组就行，每当一个点入队一次，就 counter[i]++，如果某一个点的 counter 大于 n（点的总数）那么就是环。

```
if (!vis[v]){
    vis[v]=1;
    q.push(v);
    cnt[v]++;
}
if (cnt[v]>n){
    lookln(cnt[v]);
    flag=1;break;
}
```

（四）差分约束

差分约束是一个很经典的模型，但是有其比较固有的特点，那就是不等关系，至于为什么差分约束和最短路可以联系到一起，就要从最短路更新的松弛操作讲起，当我们寻找最短路时，更新的条件是：if (d[v]>d[u]+g[u][v]) d[v]=d[u]+g[u][v]。而根据题目中的不等式，往往可以总结出一个或者几个不等式满足 f[i]-f[j]>c，这个移项之后就变成了最短路更新的形式了，所以可以利用最短路灵活的求解，至于还有什么变形，题刷的不多，就不多说了，至于比较简单的差分约束，可以看看 POJ 3169，这道题会了，应该就理解何为差分约束了，至于算法的话，还是 spfa 比较好，因为那个松弛操作可以正好对应不等式。

（五）多源最短路

也不知道表述的准确与否，就是不止一条最短路，可能是多个点到一个点的每条最短路，或者是起点到每个点的最短路，这些题目在建图方面可能有一些困难，可能有的还要建反向边，难度相对适中，只要将图建出来，那么 A 掉应该没有问题，另外就是建正向反向两个图时，我个人觉得，分开建比较好，个人习惯。

（六）最短路的一些综合和较复杂的建模（QAQ 被虐哭）

这个实在是太蛋疼了，我就说几个我做的吧，总结一下。

1）POJ 1062 昂贵的聘礼

这个题意就比较难懂，其中的意思就是在一个交易圈中，等级最高的和等级最低的人之间不能超过一个数字。好吧，就算理解了，然并卵。。。。

建图的时候第一个问题就是，怎么把价格和每个人链接起来，一看是的想法是吧价格当做点，但是这样很难表示其中的折扣价格。但如果把价格当做边，人当做点的话就可以了，之后又有一个问题，就是对于是否经过这个人来达到另一个来做决策，因为某一个物品是可以直接买的，这个有一个技巧就是另建一个源点，与每一个物品相连，权值为物品的不打折的价格。这样就可以表示不打折的物品情况了，原题也就便曾了求一条从源点到某一个点的最短路。

像这种虽然不是图的模型，但是可以转化成图的问题，关键就是找到其中两个活多个变量之间的关联方式，一帮像类似于两个变量可以通过另一个变量互相转化的关系是非常可能转化成图的模型来求解的。所以不妨把那些变量转换成点线的关联来尝试。

2）HDU 4725 The Shortest Path in Nya Graph

这道题目题意不难理解，建图方面则需要用到一个很好地技巧-分层图。在某些问题中，可能一个节点会有多个状态，而你要求的最终状态是不确定，也就是说，你不知道达到最优解时的状态，那么这个时候，分层图可以很好地解决这些问题。对每一个节点的每个状态都建立一个节点，与相关的节点连边，这样就可以保证考虑到每一个情况。但在建分层图是要注意的是同一个节点不同状态之间的限制，例如不能重复选，或者是选了这个节点之后会对后续选这个节点有影响之类的，还有就是每一层之间的关系，不过这个一般来说考虑的较少，具体情况要具体分析。

而这道题正好包含了层之间的关系，层内点的关系。首先，对于层与层之间的关系，因为相邻层之间的点可以互相到达，所以要把相邻层有点的链接，可能很多人会把所有相邻层连起来，这是不对的，因为题目的意思是从上一层的点转移到下一层的点，而不是转移到层哪儿点上去。

```
for (int i=n+1;i<n+n;i++){
    if (is[i] && is[i+1]) {
        add_edge(i,i+1,c);
        add_edge(i+1,i,c);
    }
}
```

这是一个细节，另外同一层的点的与代表该层的点之间的边的权值是 0，这样就把层和层内点连起来了，具体细节还是做了这道题再去体会吧。

3）HDU 3416 Marriage Match IV

此题，最短路再加最大流，是不是很蛋疼？

理解了题意之后，就知道，题目让求的就是最短路的条数，限制就是每一个点只能选一次。这道题乍一看很简单，仔细一看有点难，再一看我去不会写，本想深搜找，但是时间复杂度太大了，所以还是放弃了，最后大神指点，可以联系到网络流，运用拆点的方法，把每一个点拆成两个点，容量为 1，然后既可以限制该点的次数了，拆点真的需要好好掌握。另外如何把最短路的便全部抽离出，就是利用最短路的性质，具体如下：

```
for (int i=1;i<=n;i++){
    for (int j=0;j<g[i].size();j++){
        int v=g[i][j].v;
        int w=g[i][j].w;
        if (d[i]+dr[v]+w==d[ed]) {
            add_edge(i,v,1);
            add_edge(v,i,0);
        }
    }
}
```

某一条边的起点到源点的最短路和终点到汇点的最短路再加上这条边的权值等于最短路，那么这条边就是最短路中的。

4）HDU 4370 0 or 1

这道题乍一看是一个数学题，其实是一个思维转换的题目。就在于能否看出所给的矩阵是一个邻接矩阵，并且将题目所给的条件转换为图论中相关的条件，具体请看我博客吧。