**网络流24题做题总结**

1. [「网络流 24 题」搭配飞行员](https://loj.ac/problem/6000)

题意：每架飞机需要两个驾驶员，一个正驾驶员和一个副驾驶员。由于种种原因，有些驾驶员不能在同一架飞机上飞行，问如何搭配驾驶员才能使出航的飞机最多。两个正驾驶员或两个副驾驶员都不能同机飞行。

解题：**二分匹配裸题**

1. [「网络流 24 题」最小路径覆盖](https://loj.ac/problem/6002)

题意：设计一个有效算法求一个有向无环图G的最小路径覆盖并且输出其最小路径的方案。

解题：利用二分匹配中最小路径覆盖数 = |G| - 最大匹配数，其最小路径的方案就是匹配方案（枚举每一条边匹配边，为其做标记若u --> v && e[i].cap <= 0，则fa[v] = u, son[u] = v，则当fa[u] != u && son[u] == u时，u就是这条路劲覆盖的最末端，递归到fa[u] == u就OK了）。

1. [「网络流 24 题」太空飞行计划](https://loj.ac/problem/6001)

题意：现已确定了一个可供选择的实验集合 E={E1,E2,⋯,Em}，和进行这些实验需要使用的全部仪器的集合 I={I1,I2,⋯,In}。实验 Ej 需要用到的仪器是 I 的子集 Rj⊆I。 配置仪器 Ik​​ 的费用为 ck​​ 美元。实验 Ej 的赞助商已同意为该实验结果支付 pj 美元。对于给定的实验和仪器配置情况，找出净收益最大的试验计划。

解题：**最大权闭合子图模板题（附加打印方案）**

收获：有 前提条件的 网络流 用 闭合子图的思路来思考

1. [「网络流 24 题」孤岛营救问题](https://loj.ac/problem/6121)

题意：迷宫的外形是一个长方形，其南北方向被划分为 n 行，东西方向被划分为 m 列， 于是整个迷宫被划分为 n×m 个单元。南北或东西方向相邻的 2 个单元之间可能互通，也可能有一扇锁着的门，或者是一堵不可逾越的墙。迷宫中有一些单元存放着钥匙，并且所有的门被分成 p 类， 打开同一类的门的钥匙相同，不同类门的钥匙不同。 大兵瑞恩被关押在 (n,m) 单元里，并已经昏迷。迷宫只有一个入口， 在 (1,1) 单元。麦克从一个单元移动到另一个 相邻单元的时间为 1，拿取所在单元的钥匙的时间以及用钥匙开门的时间可忽略不计。 试设计一个算法，帮助麦克以最快的方式到达瑞恩所在单元，营救大兵瑞恩。

分析：将获得钥匙的状态压成二进制，对每一个状态建一层图，跑**分层图最短路。**

1. [「网络流 24 题」魔术球](https://loj.ac/problem/6003)

题意：假设有 n 根柱子，按下述规则在这 n 根柱子中依次放入编号为 1,2,3,4,⋯ 的球：每次只能在某根柱子的最上面放球；在同一根柱子中，任何 2 个相邻球的编号之和为完全平方数。 试计算出在 n 根柱子上最多能放多少个球。

解题：最小路径覆盖（不重复点）; 很显然这每一个柱子上的球组成的就是一条满足题意的路径，而题目限制了在放置好每个 x 之前，需先放置好 1,2,3,⋯x−1 的球，所以对于每一个x 所形成的图的最小路径覆盖数一定 <= n; 所以枚举X就好了，如何建图？和求最小路径覆盖的套路一样就行，第一种方法就是二分答案X，根据不同的X来进行建图，第二种方法，运用动态加边的思想（逐步加边，边逐步累加最大匹配的数量）

收获：在残留网络上**动态加边对增广路的影响**，一定要注意加边时对原图的影响。

1. [「网络流 24 题」餐巾计划](https://loj.ac/problem/6008)

题意：一个餐厅在相继的 N天里,每天需用的餐巾数不尽相同。假设第 i 天需要 ri块餐巾( i=1,2,...,N)。餐厅可以购买新的餐巾,每块餐巾的费用为 p 分;或者把旧餐巾送到快洗部,洗一块需 m 天,其费用为 f 分;或者送到慢洗部,洗一块需 n(n>m),其费用为 s 分(s<f)。每天结束时,餐厅必须决定将多少块脏的餐巾送到快洗部,多少块餐巾送到慢洗部,以及多少块保存起来延期送洗。但是每天洗好的餐巾和购买的新餐巾数之和,要满足当天的需求量。试设计一个算法为餐厅合理地安排好 N天中餐巾使用计划,使总的花费最小。编程找出一个最佳餐巾使用计划。

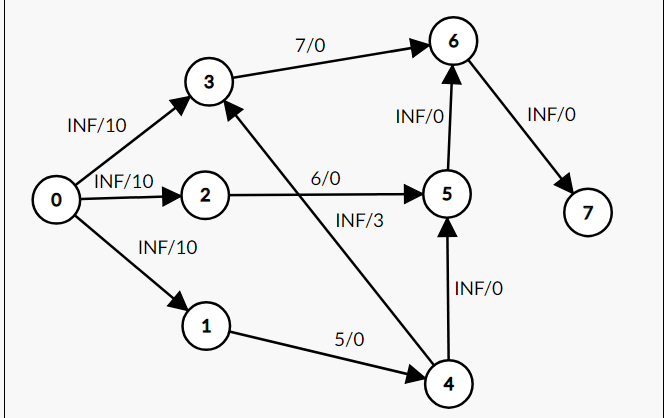


思路：

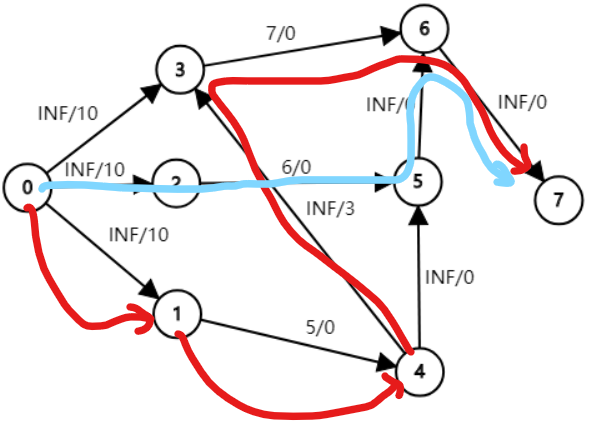
显然对于每一天的餐巾 x 来说有两个状态，从 何处来 x, 去往何处 x’

何处来：1、新买 即(addedge(S, x, INF, p))，2、从 y 快洗而来 即addedge（y’，x，INF，f），3、从z慢洗而来，即addedge（z’，x，INF，s）；去往何处：

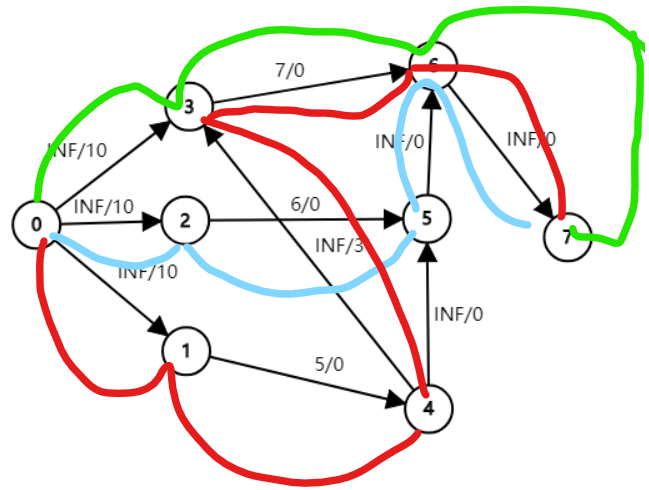
1、送去慢洗，2、送去快洗，3、留到下一天洗即addedge（x’，x’+1，INF, 0）5、最后一天的时候直接扔掉 addedge(n’，T, INF, 0)，故按此建样例的图：



理想中最大流：



而实际上呢？



为什么会这样呢？因为这样才是真正的最大流,满足题意的情况下最大流就应该等于sum(r[i])！也就是说像这样建图，最后的费用一定是 p \* sum(r[i])。

为什么会导致这样呢？红色路径 3 ---> 6 这段被流了两次，即是说 这5条餐巾 被使用过两次，但它只代表了流经红色路径 5 的流量，而我们想要的是 10 的流量（因为绿色也使用到了这 5 条餐巾），也就是说，如果有一些餐巾用了多天，最终的流量就会小于sum(r)。

**那有没有什么办法，让一条用了两天的餐巾代表2的流量呢？**这就是本题构图的巧妙之处。我们不妨先让每天开始时得到的r[i]条干净的餐巾（左边一列的节点）流向t，然后再在每天结束时从s补回r[i]条脏的餐巾（从s向右边一列的节点连r[i],0的边），得到新图的链接方式：

1. s -> i (r,p) 每天早晨可以买最多r条新餐巾 一条p分

2. s -> i' (r,0) 每天用剩下r条脏餐巾 没有代价

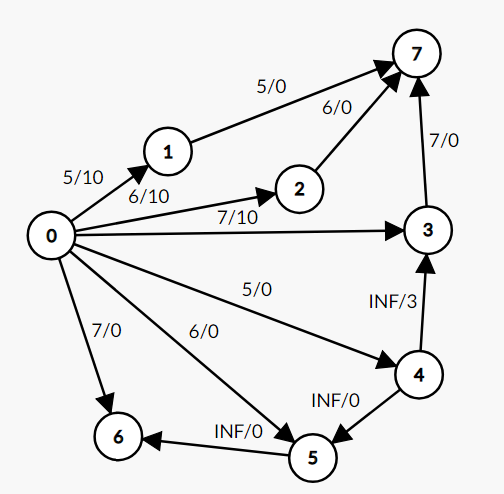
3. i -> t (r,0) 每天要用r条干净餐巾 没有代价

4. i' -> i+m (inf,f)脏毛巾送到快洗店 洗干净送回来是第i+m天每条花费代价f分

5. i' ->i+n (inf,s)脏毛巾送到慢洗店 洗干净送回来是第i+n天 每条花费代价s分

6. i' -> (i+1)' (inf,s) 每条脏毛巾留到第二天再处理 没有代价

得到新图：



收获：

1、最重要的是 **那有没有什么办法，让一条用了两天的餐巾代表2的流量呢？**

这个转换才是这个题最巧妙的地方。

2、费用流一定是建立在最大流的基础之上的，而**最大流通常都是一些约束条件，在最大流的网络中一定要搞清楚，S，T的意义，以及每一个点的流入流出情况，以及各种约束条件。**

3、就是对于拆点 X 变为 X 和X’，当X 和 X’之间连边通常是起**限制作用**，~~当 X 和 X’不连边的时候（就是连起来没啥用的时候）（鬼扯的）~~，X 和 X’不连起来是另外一种思考的方法.

1. [「网络流 24 题」圆桌聚餐](https://loj.ac/problem/6004)

题意：：假设有来自 n 个不同单位的代表参加一次国际会议。每个单位的代表数分别为 ri ​​ 。会议餐厅共有 m 张餐桌，每张餐桌可容纳 ci ​​ 个代表就餐。 为了使代表们充分交流，希望从同一个单位来的代表不在同一个餐桌就餐。 试给出满足要求的代表就餐方案。

思路：从 S 向每个单位连一条容量为人数的边，从餐桌向 T 连接一条容量为餐桌容量的边，从单位向每个餐桌连一条容量为 1 的边。直接跑最大流求解。

1. [「网络流 24 题」方格取数](https://loj.ac/problem/6007)

题意：在一个有 m×n 个方格的棋盘中，每个方格中有一个正整数。 现要从方格中取数，使任意 2 个数所在方格没有公共边，且取出的数的总和最大。试设计一个满足要求的取数算法。

分析：**最大点权独立集**。与二分图处理最大独立集的方法相似，本题还有一个点化边的思想，先将棋盘黑白染色，从 S 向每个黑点连一条容量为黑点数值的边（点化边），从白点向 T 连接一条容量为白点数值的边，从每个黑点向白点连一条容量为 inf 的边。Ans = 总价值-最小割。

收获：**最小点权覆盖 与 最大点权独立集互补**，

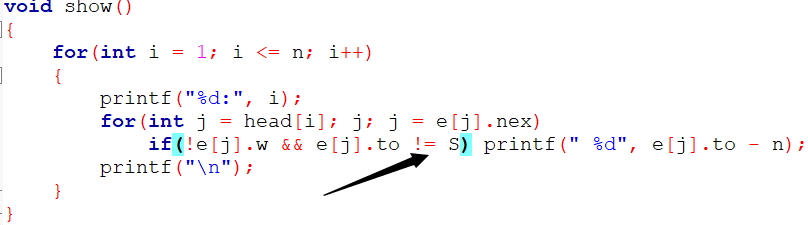
且 **最小点权覆盖集 = 最小割集(去掉 S,T)，**

**最大点权独立集 = 顶点集（去掉S,T）- 最小割集**

1. [「网络流 24 题」试题库](https://loj.ac/problem/6006)

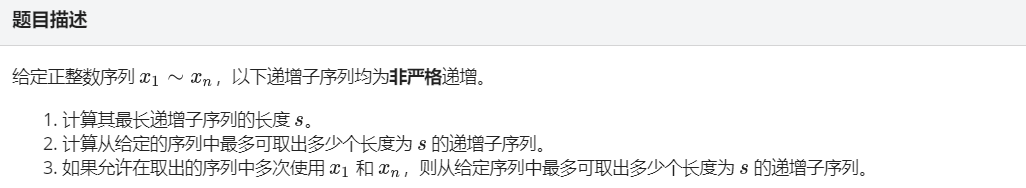
题意：假设一个试题库中有 n 道试题。每道试题都标明了所属类别。同一道题可能有多个类别属性。现要从题库中抽取 m 道题组成试卷。并要求试卷包含指定类型的试题。试设计一个满足要求的组卷算法。

分析：从 S 向每种类别连一条容量为需求的边，从题目向 T 连接一条容量为 1 的边，从每个类别向题目连一条容量为 1 的边。直接跑最大流，并输出方案。



1. [「网络流 24 题」最长递增子序列](https://loj.ac/problem/6005)

题意：



ps: xi – xn 这一对不能无限取（比如当长度为 2 时， x1 <= xn, 这时候对于第3问， 它的贡献为 1而不是无穷）

分析：令 dp[i] 表示以第 i 位开头的最长递增子序列长度，可用dp求解第一问。若 dp[i] = s，则从 S 向 i 连一条容量为 1 的边；若 dp[i] = 1 ， 则从 i 向 T 连一条容量为 1 的边；为了限定每个点只选一次，拆点，连一条容量为 1 的边；若 i<j 且 xi<=xj 且 dp[i] = dp[j] + 1，则从 i 向 j 连一条容量为 1 的边。直接跑最大流求解即可。对于第三问，把对应边有1，n的边限制放成 inf ，再跑一次最大流。

收获：**这是一个DP套网络流，通过先跑DP，处理出网络流不能处理的，在用其结果，在来建图**

1. [「网络流 24 题」骑士共存问题](https://loj.ac/problem/6226)

题意：在一个 n×n 个方格的国际象棋棋盘上，马（骑士）可以攻击的棋盘方格为“日”字。棋盘上某些方格设置了障碍，骑士不得进入。对于给定的 n×n 个方格的国际象棋棋盘和障碍标志，计算棋盘上最多可以放置多少个骑士，使得它们彼此互不攻击。

分析：二分图最大独立集，将棋盘黑白染色，从 S 向每一个可用的黑格连一条容量为 1 的边，从每一个可用的白格向 T 连一条容量为 1 的边，每一个黑格向可以攻击的白格连一条容量为 inf 的边。答案为最大独立集 = 总的可用格数-最大流。

1. [「网络流 24 题」数字梯形](https://loj.ac/problem/6010)

题意：给定一个由 n 行数字组成的数字梯形。梯形的第一行有 m 个数字。从梯形的顶部的 m 个数字开始，在每个数字处可以沿左下或右下方向移动，形成一条从梯形的顶至底的路径。 分别遵守以下规则：

1、从梯形的顶至底的 m 条路径互不相交；

2、从梯形的顶至底的 m 条路径仅在数字结点处相交；

3、从梯形的顶至底的 m 条路径允许在数字结点相交或边相交。

将按照三个规则计算出的最大数字总和输出。

分析：从 S 向顶部的每一个点连一条容量为 1 费用为 0 的边

规则一：拆点之后跑费用流（以限制每一个点只被访问过一次）；

规则二：不拆点（最后一行向T引一条容量为INF， 费用为 0的边），跑费用流；

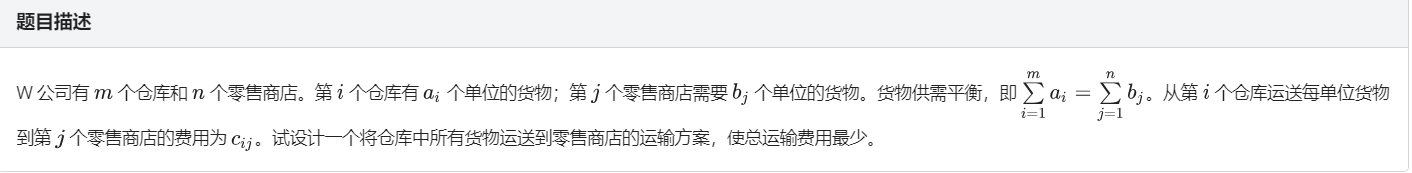
规则三：把边的容量改为 INF 。

1. [「网络流 24 题」分配问题](https://loj.ac/problem/6012)

题意：有 n 件工作要分配给 n 个人做。第 i 个人做第 j 件工作产生的效益为 cij ​​ 。试设计一个将 n 件工作分配给 n 个人做的分配方案，使产生的总效益最大。

分析：二分图最大权匹配，常规建图跑费用流即可。

1. [「网络流 24 题」运输问题](https://loj.ac/problem/6011)



分析：从 S 向仓库连一条容量为 ai 费用为 0 的边；从商店向 T 连一条容量为 bj 费用为 0 的边；从仓库向商店连一条容量为 INF 费用为 cij 的边，跑费用流即可。

1. [「网络流 24 题」负载平衡](「网络流%2024%20题」负载平衡)

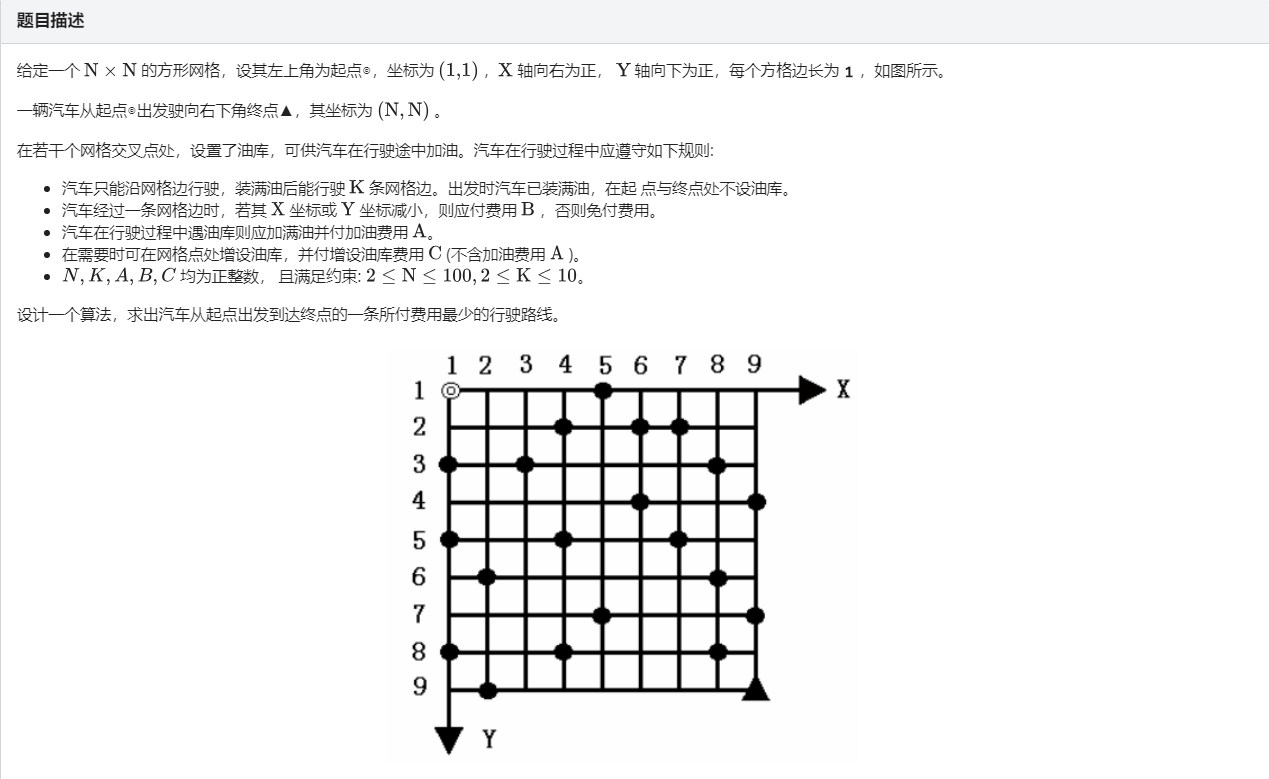
题意：公司有 n 个沿铁路运输线环形排列的仓库，每个仓库存储的货物数量为a[i]。如何用最少搬运量可以使 n 个仓库的库存数量相同。搬运货物时，只能在相邻的仓库之间搬运。

分析：就是求将 a[i] -> sum(a[i])/n 的最小费用。虽然只能在相邻仓库之间转移，但完全可以转化为任意两点都可以互相连通，只需要更改一下两点之间的最小费用 w。而且肯定是从 点A（a[A] > sum / n）向 点B(a[B] < sum / n) 进行运输，故得到建图方式，将 S ---> A ，流量为a[A] – sum / n，费用为 0；

将 B ---> T 流量为sum / n – a[B]， A ---> B 流量为INF，费用为最小费用w.

用 O(n ^ 2) 建一张完全图，再跑一次最小费用最大流就OK了。

1. [「网络流 24 题」汽车加油行驶问题](https://loj.ac/problem/6223)



K = 10, A = 2

B = 3, C = 6

Ans = 12

分析：分层图最短路。

收获：对于这种 对这些 对 步数有限制的，而且限制条件也很小的，最优化问题，可以考虑 分层图最短路来求，用分层的 来满足 步数限制，因为每一个步数限制就是一种不同的状态，也即是每一层是一种不同的状态。

1. [「网络流 24 题」航空路线问题](https://loj.ac/problem/6122)

题意：给定一张航空图，图中顶点代表城市，边代表两个城市间的直通航线。现要求找出一条满足下述限制条件的且途经城市最多的旅行路线：从最西端城市出发，单向从西向东途经若干城市到达最东端城市，然后再单向从东向西飞回起点（可途经若干城市）；除起点城市外，任何城市只能访问一次。对于给定的航空图，试设计一个算法找出一条满足要求的最佳航空旅行路线。

分析：题目是从 最西 到 最东，然后在返回！那么就可以转化为 **求两条不相交的路径且路径之和最长**，

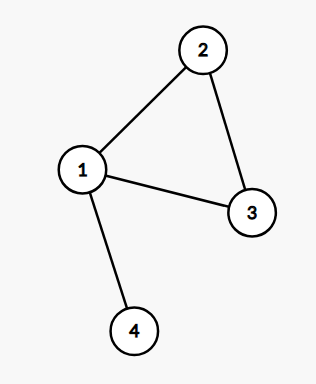
首先对于某个点拆成入点和出点，然后入点与出点连接容量为1，费用为0的边。

对于每两个相连的城市，从出发的出点向到达的入点连接容量为1，费用为1的边。

由于起点和终点要经过两次，所以这两个点拆出的入点向出点连接的边容量为2。

坑点：

如给出如下航空图：



故如果有1到n的边，是可以直接往返的，所以有这种边，容量也为2。

然后跑最大费用最大流。就OK

收获：在路径输出时，可以采用 “回流”的方法：（**其中e[i].ww 为 e[i].w的初始值**（没有参与网络流之前的值））

1. **void show(int u, int flag)**
2. **{**
3. **for(int i = head[u]; i; i = e[i].nex)**
4. **{**
5. **int v = e[i].to;**
6. **if(e[i].w < e[i].ww && e[i].ww)**
7. **{**
8. **show(v, flag);**
9. **++e[i].w;**
10. **break;**
11. **}**
12. **}**
13. **}**
14. [「网络流 24 题」星际转移](https://loj.ac/problem/6015)

题意：现有 n 个太空站位于地球与月球之间，且有 m 艘公共交通太空船在其间来回穿梭。每个太空站可容纳无限多的人，而每艘太空船 i 只可容纳 Hi ​​ 个人。每艘太空船将周期性地停靠一系列的太空站，例如：1,3,4 表示该太空船将周期性地停靠太空站 134134134⋯ 每一艘太空船从一个太空站驶往任一太空站耗时均为 1。人们只能在太空船停靠太空站（或月球、地球）时上、下船。 初始时所有人全在地球上，太空船全在初始站。试设计一个算法，找出让所有人尽快地全部转移到月球上的运输方案。

分析：这个题其实和“魔术球”这个题类似。题意就相当于一些人要从 A ---> B ，空间站为中转点，通过飞船转移，这些飞船每一天出现在固定空间站，也即是说天数不一定的时候，整个运输图就会发生变化，而每一天之间又是有关系的。就好像，每一天的空间站是不一样的（因为飞船飞行线路不一样了），所以考虑到 地球（0号点）、n个空间站和月亮（n + 1号点）共 n + 2 个点，每一天分裂一次（即拥有一个新的状态），有因为共经历 t 天，完成运输，故 将要分裂出 t \* (n + 2) 个点【即是说将每一天的这 n + 2个点看成一个新的点】，枚举每一天day，令yesterday = (n + 2) \* (day – 1)，today = (n + 2) \* day；则每一个点 i’ = yesterday + i（代表昨天对应的点）, i = today + i（代表今天的点）；故 i’--->i，容量为INF， (n + 1) + today ---> T，容量为 INF;

在从每一个飞船前一天所在的太空站连向后一天的太空站，流量为飞船可容纳人数；

在最开始的时候 链接一条， S ----> 0，容量为 INF 的边

跑最大流直到不小于总人数即可。（每一天的最大流应累加起来）

收获：又收获一种建图方式，**利用分层图思维建图**。

类中定义的数组，可能在使用的时候会出现异常（通常表现在类所占用的空间资源较多时），可以把数组定义在全局。

1. [「网络流 24 题」火星探险问题](https://loj.ac/problem/6225)

题意： 登陆舱着陆后，探测车将离开登陆舱向先期到达的传送器方向移动。 探测车在移动中还必须采集岩石标本，每一块岩石标本由最先遇到它的探测车完成采集。 每块岩石标本只能被采集一次。 岩石标本被采集后，其他探测车可以从原来岩石标本所在处通过。 探测车不能通过有障碍的地面。 本题限定探测车只能从登陆处沿着向南或向东的方向朝传送器移动，而且多个探测车可以在同一时间占据同一位置。 如果某个探测车在到达传送器以前不能继续前进，则该车所采集的岩石标本将全部损失。 用一个 P×Q 网格表示登陆舱与传送器之间的位置。登陆舱的位置在 (X1,Y1) 处，传送器 的位置在 (XP,YQ) 处。 给定每个位置的状态，计算探测车的最优移动方案，使到达传送器的探测车的数量最多， 而且探测车采集到的岩石标本的数量最多。

分析：将原图中的每一个顶点拆成两个点：若其为岩石顶点，从 i 向 i′ 连一条容量为 1 费用为 1 的边；若其为非障碍点，从 i 向 i′ 连一条容量为 inf 费用为 0 的边；对于相邻的点 j ，从 i′ 向 j 连一条容量为 inf 费用为 0 的边。从 S 向 (X1,Y1) 连一条容量为探测车数量费用为 0 的边；从 (XP,YQ) 向 T 连一条容量为探测车数量费用为 0 的边。跑最大费用最大流即可。

“每个石块点的价值只能计算一次，所以容量限制要设为1，

“多个探测车可以在同一时间占据同一位置”，非障碍点内部要有一条容量为无穷大的边。”

收获：**反向边在输输出方案时的也是有作用的啊！**

1. **void show(int u, int x)**
2. **{**
3. **if(u == id[n][m]) return;**
4. **for(int i = head[u + base]; i; i = e[i].nex)**
5. **{**

反向边的head 一定是奇数

流过反向边的流量一定等于流过正向边的流量。

1. **if(i&1 || !e[i ^ 1].w) continue;**
2. **int v = e[i].to;**
3. **printf("%d %d\n", x, getTurn(u, v));**
4. **e[i ^ 1].w--; e[i].w ++;**
5. **show(v, x);**
6. **return;**
7. **}**
8. }
9. [「网络流 24 题」深海机器人问题](https://loj.ac/problem/6224)

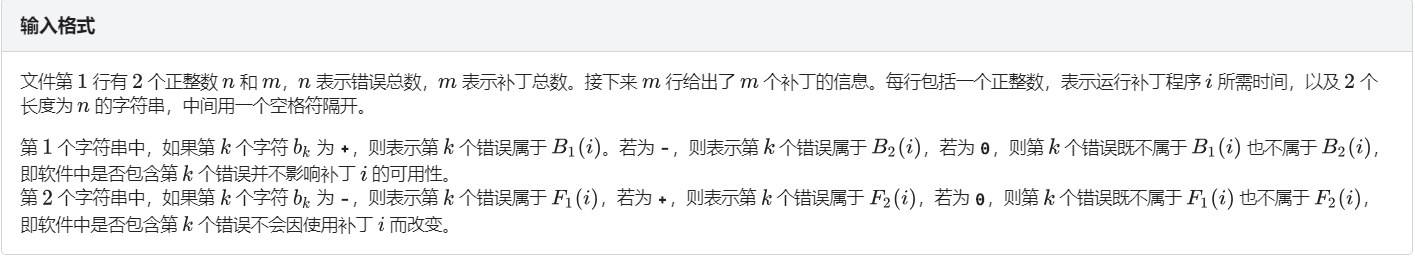
题意：潜艇内有多个深海机器人。潜艇到达深海海底后，深海机器人将离开潜艇向预定目标移动。 深海机器人在移动中还必须沿途采集海底生物标本，沿途生物标本由最先遇到它的深海机器人完成采集。 每条预定路径上的生物标本的价值是已知的，而且生物标本只能被采集一次。 本题限定深海机器人只能从其出发位置沿着向北或向东的方向移动，而且多个深海机器人可以在同一时间占据同一位置。 用一个 P×Q 网格表示深海机器人的可移动位置。西南角的坐标为 (0,0) ，东北角的坐标为 (Q,P) 。 给定每个深海机器人的出发位置和目标位置，以及每条网格边上生物标本的价值。 计算深海机器人的最优移动方案， 使深海机器人到达目的地后，采集到的生物标本的总价值最高。

分析：每条边的贡献只能计算一次，考虑拆边：一条边容量为 1 费用为价值，一条边容量为 inf 费用为 0 。跑最小费用最大流即可。

收获：**每条边的贡献只能计算一次，考虑拆边，拆为一条仅经过的边和一条有着这条边自身属性 的边。**

1. [「网络流 24 题」软件补丁](https://loj.ac/problem/6009)

题意：某公司发现其研制的一个软件中有 n 个错误，随即为该软件发放了一批共 m 个补丁程序。对于每一个补丁 i ，都有 2 个与之相应的错误集合 B1(i) 和 B2(i) ，使得仅当软件包含 B1(i) 中的所有错误，而不包含 B2(i) 中的任何错误时，才可以使用补丁 i。补丁 i 将修复软件中的某些错误 F1(i) ，而同时加入另一些错误 F2(i)。另外，每个补丁都耗费一定的时间。 试设计一个算法，利用公司提供的 m 个补丁程序将原软件修复成一个没有错误的软件，并使修复后的软件耗时最少。



分析：假设000…000为软件的初始状态（即n个bug都处于没有修好的状态），111…111为最终状态（n个bug都修好了），求最小花费。想到用二进制来压缩状态。令

B1[i] : 补丁i包含的错误集合，故 B1[i] **|=** s1[j] == ‘+‘ ? (1<<j) : 0；

B2[i]：补丁i不能有的错误集合，故（B2[i] **|=** s1[j] == ‘-‘ ? (1 << j) : 0；当然B2[i] 也表示 送来 让 i处理的软件必须已经被修好的集合

F1[i]：安装补丁i后，新增bug的补集，故F1[i] **|=** s2[j] != ‘+’ ? (1<<j) : 0；

F2[i]：安装补丁i后，被修好的集合，故 F2[i] |= s2[j] == ‘-‘ ? (1<<j) : 0；

在考虑 对于 任意 一个转态 u，是否能通过 补丁i ？

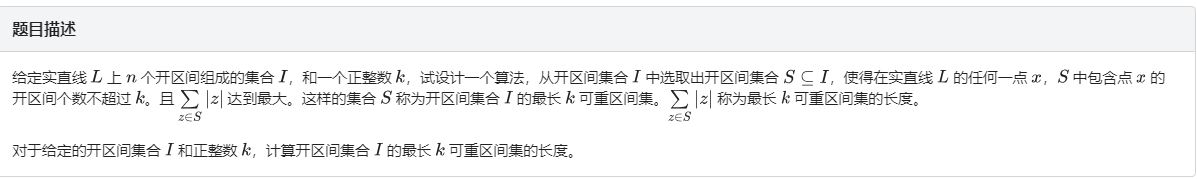
若 B1[i] & u != 0，则说明 u 中 至少少了一位 B1所需要错误。

若 B2[i] & u != B2[i]，则说明 u 中 至少有一个bug没有被修好。

那 u通过 i被转化成的v为？V = (u | d[i]) & c[i]；最后再用SPFA跑一个最短路。

1. [「网络流 24 题」最长 k 可重区间集](https://loj.ac/problem/6014)

题意：



分析：两种建图方式：

法一：与“十七、航空路线问题”建图方式类似。这个题也可转化为：从最左端出发到达最右端，每条路径最多被经过K次，当然这里的最左端，不是1号线段了，而是源点S，为了保证 所有路径至多被经过K次（不就是流量限制？）将源点S也进行拆点，拆为 S、S’。具体方法如下：

按左端点排序所有区间，把每个区间拆分看做两个顶点<i.a><i.b>，建立附加源S汇T，以及附加顶点S'。

1、连接S到S'一条容量为K，费用为0的有向边。

2、从S'到每个<i.a>连接一条容量为1，费用为0的有向边。

3、从每个<i.b>到T连接一条容量为1，费用为0的有向边。

4、从每个顶点<i.a>到<i.b>连接一条容量为1，费用为区间长度的有向边。

5、对于每个区间i，与它右边的不相交的所有区间j各连一条容量为1，费用为0的有向边。

这种方法是 **最大权不相交路径** 的一般解题思路，建图的复杂为 O(N^2)

法二：我们改变一下思路，把端点作为网络中的顶点，区间恰恰是特定一些端点之间的边，这样建模的复杂度更小。

离散化所有区间的端点，把每个端点看做一个顶点，建立附加源S汇T。

1、从S到顶点1（最左边顶点）连接一条容量为K，费用为0的有向边。

2、从顶点2N（最右边顶点）到T连接一条容量为K，费用为0的有向边。

3、从顶点i到顶点i+1(i+1<=2N)，连接一条容量为无穷大，费用为0的有向边。

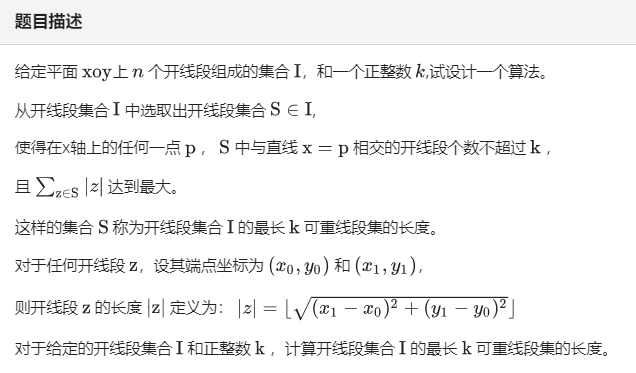
4、对于每个区间[a,b]，从a对应的顶点i到b对应的顶点j连接一条容量为1，费用为区间长度的有向边。

这种建图的方法 建图的复杂度为 O（N）

收获：求 **K条最大权不相交路径** 的一般建图方法和一种特殊的技巧建图方法，以及将**源点 S** 进行拆点限流的思想。

1. [「网络流 24 题」最长 k 可重线段集问题](https://loj.ac/problem/6227)

题意：



分析：将 二维平面的 点投影在 X 轴上（毕竟与纵坐标好像真的没什么关系），然后，就变成了 上一题。就是要注意 垂直与X轴的线段，重合的线段，和 既和X轴垂直又和其他线段相交 的线段。

1. 总结

“网络流判定问题更适合枚举答案，而不是二分，因为新增一些点和边只需要在原有的基础上增广，不必重新求网络流。

建立供求网络，把二分图X集合中所有节点看做供应节点，Y集合所有节点看做需求节点，在能一次搬运满足供需的Xi和Yj之间连接一条费用为1的有向边，表示搬运一个单位货物费用为1。另外还要在Xi与相邻的Xj之间连接边，表示货物可以暂时搬运过去，不立即满足需求，费用也为1。最大流满足了所有的盈余和亏损供求平衡，最小费用就是最少搬运量。

一些建图技巧：

1、拆点。

2、拆边。

3、分层图思想。

4、二进制状态压缩。

一些扩展题型：

1、最大权独立集，最小权覆盖集。

2、最大密度子图。

3、最长不相交路径。

4、供求网络。