预览输出

题目名称	硬币
题目类型	传统型
目录	`coin`
可执行文件名	`coin`
输入文件名	`coin.in`
输出文件名	`coin.out`
每个测试点时限	1.0 秒
内存限制	512 MiB
子任务数目	10
测试点是否等分	是

提交源程序文件名

对于 C++ 语言	`coin.cpp`
对于 C 语言	`coin.c`
对于 Pascal 语言	`coin.pas`

编译选项

对于 C++ 语言	`-O2 -std=c++14`
对于 C 语言	`-O2 -std=c11`
对于 Pascal 语言	`-O2`

硬币（coin）

Author: yuxaij@gmail.com

【证明存在大小为 f 整数流】

先考虑 f 不为整数的情况。假设有 $flow(F)=f$，不妨令 $E'=\{F(e)\notin\mathbb{N}^*\}$。容易证明，$G(V,E')$ 中除去 S,T 外不存在度数为 1 的点。从 S 开始在 $G(V,E')$ 上做 DFS，则要么必然找到一个环，要么找到一条 $S\to T$ 的路径。

对于后者，由于 $S\to T$ 路径上所有边的剩余容量不为 0，因此可以找到剩余容量 rc 最大的边，将该路径上所有边的流量加上这个剩余容量 rc 。若 $rc\geq\lceil f\rceil-f$，则显然可以将 f 调整为整数大小的流。对于前者，类似地找到环上剩余容量 rc 最大的边，将该环上所有边的流量加上 rc。

重复以上操作，由于每次操作不会使 E' 以外的边流量改变，而至少将某一条 $e\in E'$ 的流量改为整数，因此 $|E'|$ 严格递减，且 $flow(F')$ 严格递增。当 $|E'|=0$ 时，显然有 $flow(F)=outflow(S)\in\mathbb{N}^*$。

若 f 为整数，则由于每个点的度都不为 1，DFS 中必然找到一个环。同理可证。

【边的容量为 1】

考虑边容量为 1 的情况。不妨假设存在一个整数解的最大流 OPT，$flow(OPT)>flow(F)$。

考虑流量平衡条件。由于 $flow(OPT)>flow(F)$，则必然存在一条边 $e(S,u)\in E$ 满足 $OPT(e)=1,F(e)=0$。根据流量平衡我们有

$$
\begin{aligned}
inFlow(OPT,u)-inFlow(F,u) &= \sum_{e(v,u)\in E} OPT(e)-F(e)\\
&= \sum_{e(u,v')\in E} OPT(e)-F(e)\\
&= outFlow(OPT,u)-outFlow(F,u)
\end{aligned}
$$

$$\Rightarrow$$

$$
\sum_{\text{e adjacent to u}} OPT(e)-F(e) = 1+\sum_{e\neq(S,u),\text{e adjacent to u}} (-1)^{[b_e=u]}(OPT(e)-F(e))
$$

$$=0$$

因此除去 e 以外，与 u 相邻的边中至少满足：

- 要么存在一条边 $e'(u,v)\in E,OPT(e')=1,F(e')=0$
- 要么存在一条边 $e'(v,u)\in E,OPT(e')=0,F(e')=1$

同理可以推出 v' 上同样存在类似的条件。重复这个步骤，直到找到一个环 C，或者找到一条到 T 的路径 P。

令 $F' = F \oplus C/P$。则对于后者，F' 满足流量平衡且是一个流量 $+1$ 的可行流。对于前者，F' 流量不变满足流量平衡，且 $E' = |\{OPT(e) \neq F(e)\}|$ 规模减小，重新从 S 出发增广。若 $E' = 0$，则此时 $F = OPT$，矛盾。

因此，最终我们总能找到一条 $S \to T$ 的路径，问题在于在这之前可能会找到若干个环。这个问题实际上可以被整理为：

> 给定一张图 $G'(V, E'_{F,OPT})$，其中 $E'_{F,OPT}$ 边集由以下方法构成：$\forall e(u, v) \in E$
>
> - 若 OPT(e) = 1, F(e) = 0，则向 E' 加入一条边 (u, v)
> - 若 OPT(e) = 0, F(e) = 1, 则向 E' 加入一条边 (v, u)
>
> 求该图中一条 $S \to T$ 的路径。这个显然直接 DFS 即可。

但是由于 OPT 是未知的，我们并不能得到对应的 $E'_{F,OPT}$。因此可以考虑如何构造一个包含 $E'_{F,OPT}$ 的边集 rE。由于 $E'_{F,OPT}$ 被包含，因此若 OPT 存在，则图 $G(V, rE)$ 中 S 必然可以访问 T，必要性得证。再考虑充分性，若我们在 $G(V, rE)$ 中找到了一条 S 访问 T 的路径，则只要保证我们所找到的这条路径必然能使 F 的流量 $+1$ 即可。

这个边集容易构造：

- 若 F(e) = 0，则将这条边加入 rE
- 若 F(e) = 1, 则将其的反向边加入 rE 中

容易证明，若找到一条从 $S \to T$ 的路径 P，则 $F \oplus P$ 满足流量平衡和容量限制，同时流量 $+1$。充分性得证。

$\triangleright G(V, rE)$ 即残量网络，P 即一条增广路。

【一般情况】

不妨将一条容量为 c 的边拆分成 c 条边。显然，两个流系统是等价的。因此，对于一条容量为 c_e，流量为 $F(e)$ 的边，其贡献了 $F(e) - c_e$ 条正向边 e，c_e 条逆向边。由于令流量 $+1$ 只需要判断 S, T 连通性，因此分别只计至多一条边即可。

若将 k 条重边合并成一条容量为 c 的边，则首先不影响正确性；其次，对于找到的路径 P，设路径上边的最小容量为 $c_{min}(P)$，则等价于我们找到了 $c_{min}(P)$ 的流量。

【最小费用 +1】

假设最优解将 F 修改为 F'，类似考虑 $F' - F$ 表示成的一个图 $G(V, E')$：

- 若 $F'(e) - F(e) > 0$，则在 E' 中连 $F'(e) - F(e)$ 条边 e，边权为 v_e
- 若 $F'(e) - F(e) < 0$，则在 E' 中连 $F(e) - F'(e)$ 条边 e 的反向边，边权为 v_e。

修改 F 到 F' 的费用 $cost(G)$ 即为图 G 中所有边的边权和。

显然，若图 $G(V, E')$ 中存在环 C，则删去这个环得到一个新的 $F' \oplus C$ 和图 $G'(V, E' - C)$，且 $flow(F') = flow(F' \oplus C)$，$cost(G') < cost(G)$。

重复该操作，直到 E' 中不存在环，即只有一条从 S 到 T 的路径 $F \oplus F'$。根据之前的结论，这条路径必然存在于 rE。

故在 rE 中由 S 向 T 跑最短路即可。

$\triangleright$ 注意这里的费用和实际的费用流有区别。