习题一

1 习题一

0.1

	$f(n)$	$g(n)$
(a)	$n-100$	$n-200$

$f = \Theta(g)$

1 习题一

0.1

	$f(n)$	$g(n)$
(b)	$n^{1/2}$	$n^{2/3}$

$$f = O(g)$$

1 习题一

	$f(n)$	$g(n)$
0.1 (c)	$100n + \log n$	$n + (\log n)^2$

$$f = \Theta(g)$$

1 习题一

0.1	$f(n)$	$g(n)$
(d)	$n \log n$	$10n \log 10n$

$$g(n) = 10n(\log n + \log 10) = 10n \log n + 10 \log 10\, n$$

$$f = \Theta(g)$$

1 习题一

0.1	$f(n)$	$g(n)$
(e)	$\log 2n$	$\log 3n$

$$f(n) = \log n + \log 2$$

$$g(n) = \log n + \log 3$$

$$f = \Theta(g)$$

1 习题一

0.1	$f(n)$	$g(n)$
(f)	$10 \log n$	$\log(n)^2$

$$g(n) = 2 \log n$$

$$f = \Theta(g)$$

1 习题一

0.1	$f(n)$	$g(n)$
(g)	$n^{1.01}$	$n\log^2 n$

$$\frac{f(n)}{g(n)} = \frac{n^{1.01}}{n\log^2 n} = \frac{n^{0.01}}{\log^2 n}$$

$$\lim_{n\to\infty}\frac{f(n)}{g(n)} = \lim_{n\to\infty}\frac{n^{0.01}}{\log^2 n} = \lim_{n\to\infty}\frac{0.01n^{-0.99}}{2\log n\cdot\frac{\ln 2}{n}} = \lim_{n\to\infty}\frac{0.01n^{0.01}}{2\ln 2\log n} = \lim_{n\to\infty}\frac{10^{-4}n^{0.01}}{2\ln^2 2} = \infty$$

$$f = \Omega(g)$$

1 习题一

0.1

	$f(n)$	$g(n)$
(h)	$n^2/\log n$	$n(\log n)^2$

$$\frac{f(n)}{g(n)} = \frac{n^2/\log n}{n(\log n)^2} = \frac{n}{(\log n)^3}$$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{n}{(\log n)^3} = \lim_{n\to\infty} \frac{1}{3(\log n)^2 \frac{\ln 2}{n}} = \lim_{n\to\infty} \frac{n}{3\ln 2\,(\log n)^2}$$

$$= \lim_{n\to\infty} \frac{n}{6\ln^2 2 \log n} = \lim_{n\to\infty} \frac{n}{6\ln^3 2} = \infty$$

$$f = \Omega(g)$$

1 习题一

0.1

	$f(n)$	$g(n)$
(i)	$n^{0.1}$	$(\log n)^{10}$

$$\frac{f(n)}{g(n)} = \frac{n^{0.1}}{(\log n)^{10}}$$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{n^{0.1}}{(\log n)^{10}} = \lim_{n\to\infty} \frac{0.1n^{-0.9}}{10(\log n)^9 \frac{\ln 2}{n}} = \lim_{n\to\infty} \frac{0.1n^{0.1}}{10 \ln 2 (\log n)^9} = \infty$$

$$f = \Omega(g)$$

1 习题一

	$f(n)$	$g(n)$
0.1 (j)	$(\log n)^{\log n}$	$n/\log n$

$$x = \log n$$

$$f(x) = x^x \quad g(x) = 2^x/x$$

$$\frac{f(x)}{g(x)} = \frac{x^{x+1}}{2^x} = \left(\frac{x}{2}\right)^x x$$

$$f = \Omega(g)$$

1 习题一

0.1	$f(n)$	$g(n)$
(k)	$\sqrt{n}$	$(\log n)^3$

$$\frac{f(n)}{g(n)} = \frac{n^{1/2}}{(\log n)^3}$$

$$f = \Omega(g)$$

1 习题一

0.1

	$f(n)$	$g(n)$
(l)	$n^{1/2}$	$5^{\log_2 n}$

$$g(n) > 2^{\log_2 n} = n$$

$$f = O(g)$$

1 习题一

0.1

	$f(n)$	$g(n)$
(m)	$n2^n$	3^n

$$\frac{f(n)}{g(n)} = \frac{n2^n}{3^n} = n\left(\frac{2}{3}\right)^n$$

$$f = O(g)$$

1 习题一

0.1

	$f(n)$	$g(n)$
(n)	2^n	2^{n+1}

$$f = \Theta(g)$$

1 习题一

0.1

	$f(n)$	$g(n)$
(o)	$n!$	2^n

$$f = \Omega(g)$$

1 习题一

	$f(n)$	$g(n)$
0.1	$(\log n)^{\log n}$	$2^{(\log_2 n)^2}$
(p)		

$$g(n) = 2^{\log_2 n \cdot \log_2 n} = n^{\log n}$$

$$f = O(g)$$

1 习题一

0.1

	$f(n)$	$g(n)$
(q)	$\sum_{i=1}^{n} i^k$	n^{k+1}

Faulhaber公式

$$\sum_{k=1}^{n} k^p = \frac{n^{p+1}}{p+1} + \frac{n^p}{2} + \sum_{k=2}^{p} \frac{B_k p!\, n^{p-k+1}}{k!\,(p-k+1)!}$$

$f = \Theta(g)$

1 习题一

0.2 c is a positive real number, $g(n) = 1 + c + c^2 + \cdots + c^n$

(a)

$$c < 1, g(n) = \frac{1 - c^{n+1}}{1 - c}$$

$$\frac{g(n)}{1} \leq \frac{1}{1 - c} \qquad \frac{g(n)}{1} \geq 1$$

$$g(n) = \Theta(1)$$

1 习题一

0.2 c is a positive real number, $g(n) = 1 + c + c^2 + \cdots + c^n$

(b)

$$c = 1, g(n) = n + 1$$

$$g(n) = \Theta(n)$$

1 习题一

0.2 c is a positive real number, $g(n) = 1 + c + c^2 + \cdots + c^n$

(c)

$$c > 1, g(n) = \frac{c^{n+1} - 1}{c - 1}$$

$$\frac{g(n)}{c^n} = \frac{c - c^{-n}}{c - 1} \leq \frac{c}{c - 1} \qquad \frac{g(n)}{c^n} = \frac{c - c^{-n}}{c - 1} \geq 1$$

$$g(n) = \Theta(c^n)$$

1 习题一

1.14 find an efficient way to compute $F_n \ mod \ p$

$$F_n = F_{n-1} + F_{n-2}$$

$$(F_n \quad F_{n-1}) = (F_{n-1} \quad F_{n-2}) \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = (F_2 \quad F_1) \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-2}$$

用矩阵快速幂求$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-2}$，计算中间结果时$mod \ p$

1 习题一

1.14 find an efficient way to compute $F_n \ mod \ p$

```
int p;
int temp[2][2];
void multiply(int a[][2], int b[][2]){
    memset(temp,0,sizeof(temp));
    for(int i=0; i<2; i++)
        for(int j=0; j<2; j++)
            for(int k=0; k<2; k++)
                temp[i][j]=(temp[i][j]+a[i][k]*b[k][j])%p;
    for(int i=0; i<2; i++)
        for(int j=0; j<2; j++)
            a[i][j]=temp[i][j];
}
```

```
int res[2][2];
void qpow_m(int a[][2],int n){
    memset(res,0,sizeof(res));
    for(int i=0;i<2;i++)
        res[i][i]=1;
    while(n){
        if(n&1)
            multiply(res,a,N);
        multiply(a,a,N);//a=a*a
        n>>=1;
    }
}
```

1 习题一

1.14 find an efficient way to compute $F_n \ mod \ p$

$$F_n = F_{n-1} + F_{n-2}$$

$$(F_n \quad F_{n-1}) = (F_{n-1} \quad F_{n-2}) \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = (F_2 \quad F_1) \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-2}$$

用矩阵快速幂求$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-2}$，计算中间结果时$mod \ p$

最后求$res_{11} + res_{21} \ mod \ p$

中间所得结果均小于p，每一步运算的复杂度为$O(\log^2 p)$，
总复杂度为$O(\log n \log^2 p)$

1 习题一

1.20

(a) find the inverse of 20 mod 79

$(20,79) = 1$，79为质数

由费马小定理得，$20^{78} \equiv 1\ (mod\ 79)$

$$20^{-1} \equiv 20^{77} \equiv 20 \times (400)^{38}$$
$$\equiv 20 \times 5^{38} \equiv 20 \times 25 \times 125^{12}$$
$$\equiv 500 \times 46^{12} \equiv 26 \times 62^{6}$$
$$\equiv 26 \times 52^{3} \equiv 8 \times 26^{4}$$
$$\equiv 8 \times 44^{2} \equiv 8 \times 40$$
$$\equiv 4\ (mod\ 79)$$

1 习题一

1.20

(b) find the inverse of 3 $mod\ 62$

$(3,62) = 1$

由欧拉定理得，$3^{\varphi(62)} \equiv 1\ (mod\ 62)$

$3^{-1} \equiv 3^{\varphi(62)-1} \equiv 3^{29}$

$\equiv 3 \times 19^7$

$\equiv 3 \times 19 \times 51^3$

$\equiv 57 \times 51 \times 59$

$\equiv (-5) \times 51 \times (-3)$

$\equiv 21\ (mod\ 62)$

1 习题一

1.20

(c) find the inverse of $21\ mod\ 91$

$(21,91) = 7 \neq 1$

$21^{-1}\ (mod\ 91)$不存在

1 习题一

1.20

(d) find the inverse of 5 mod 23

$(5,23) = 1$

设$5^{-1} \equiv x\ (mod\ 23) \Rightarrow 5x \equiv 1\ (mod\ 23) \Rightarrow 5x + 23y = 1$

由扩展欧几里得算法，得

$5x_1 + 23y_1 = 1$	$(x_5, y_5) = (0,1)$
$\Rightarrow 3x_2 + 5y_2 = 1$	$\Rightarrow (x_4, y_4) = (1,0)$
$\Rightarrow 2x_3 + 3y_3 = 1$	$\Rightarrow (x_3, y_3) = (-1,1)$
$\Rightarrow x_4 + 2y_4 = 1$	$\Rightarrow (x_2, y_2) = (2,-1)$
$\Rightarrow 0x_5 + y_5 = 1$	$\Rightarrow (x_1, y_1) = (-9,2)$

$5^{-1} \equiv -9 \equiv 14\ (mod\ 23)$

1 习题一

1.31

(a) If N is an n-bit number, how many bits long is N!

$N!$的位数为$\log N! = \Theta(N \log N)$

思路一

$$N! \leq N^N, \log N! = O(\log N^N) = O(N \log N)$$

$$N! \geq \frac{N}{2} \cdot \left(\frac{N}{2} + 1\right) \cdots N \geq \frac{N}{2}^{\frac{N}{2}}, \log N!$$

$$= \Omega\left(\log \frac{N}{2}^{\frac{N}{2}}\right) = \Omega\left(\frac{N}{2} \log \frac{N}{2}\right) = \Omega(N \log N)$$

1 习题一

1.31

(a) If N is an n-bit number, how many bits long is N!

$N!$的位数为$\log N! = \Theta(N \log N)$

思路二

$$\log N! = \sum_{i=1}^{N} \log i = \sum_{i=1}^{\log N - 1} \left(2^{i+1} - 2^{i}\right) i + \left(N - 2^{\log N}\right) \log N$$

$$= (\log N - 2) 2^{\log N} + 2 + N \log N - \log N \, 2^{\log N}$$

$$= N \log N - 2^{\log N + 1} + 2 = \Theta(N \log N)$$

1 习题一

1.31

(a) If N is an n-bit number, how many bits long is N!

$N!$的位数为$\log N! = \Theta(N \log N)$

思路三

由斯特林公式，得$N! = \sqrt{2\pi N}\left(\frac{N}{e}\right)^N$

$$\log N! = \log \sqrt{2\pi N}\left(\frac{N}{e}\right)^N$$

$$= \log\sqrt{2\pi} + \frac{1}{2}\log N + N\log N - N\log e$$

$$= \Theta(N \log N)$$

1 习题一

1.31

(b) Give an algorithm to compute N! and analyze its running time

直接从1连续乘到N

考虑第i步，即$(i-1)!\cdot i$

$(i-1)!$为$\Theta\big((i-1)\log(i-1)\big)=\Theta(i\log i)$位，$i$为$\log i$位，则第$i$步的复杂度为$\mathrm{O}(i\log^2 i)$

整个算法的复杂度为

$$\mathrm{O}\left(\sum_{i=1}^{N} i\log^2 i\right)=\mathrm{O}\left(\log^2 N\sum_{i=1}^{N} i\right)=\mathrm{O}(N^2\log^2 N)$$

1 习题一

1.35

(a) If p is prime, then we know every number $1 \le x < p$ is invertible modulo p. Which of these numbers are their own inverse?

$$x^{-1} \equiv x \,(mod\ p) \Rightarrow x^2 \equiv 1 \,(mod\ p) \Rightarrow x^2 = kp + 1$$

若$k = 0$，$x^2 = 1$，$x = 1$

若$k \neq 0$，$kp = x^2 - 1 = (x-1)(x+1)$，则$p|(x-1)$或$p|(x+1)$

若$p|(x-1)$，$0 \le x - 1 \le p - 2$，不成立

若$p|(x+1)$，$2 \le x + 1 \le p$，$x = p - 1$

1 习题一

1.35

(b) By pairing up multiplicative inverses, show that $(p-1)! \equiv -1 \ (mod\ p)$ for prime p.

$(p-1)! = 1 \times 2 \times \cdots \times (p-1)$，$p$为素数，则$1,2,\cdots,p-1$共有偶数个数

考察$x \in \{2,3,\cdots,p-2\}$，则由(a)可知$x^{-1} \not\equiv x \ (mod\ p)$，则$\exists y \in \{2,3,\cdots,p-2\}$且$y \neq x$，$x^{-1} \equiv y \ (mod\ p)$，同理$y^{-1} \equiv x \ (mod\ p)$，$xy \equiv 1 \ (mod\ p)$

则$2,3,\cdots,p-2$这$p-3$个偶数可两两配对组成$\frac{p-3}{2}$个互逆对

$(p-1)! \equiv 1 \times (p-1) \equiv -1 \ (mod\ p)$

1 习题一

1.35

(c) Show that if N is not prime, then $(N-1)! \not\equiv -1 \ (mod\ N)$

思路一：反证

假设存在合数N且$(N-1)! \equiv -1 \ (mod\ N)$
则$(N-1)! \times (N-1) \equiv 1 (mod\ N)$，$(N-1)!^{-1}$为$N-1$，
即$(N-1)!\ mod\ N$的逆元存在，则$gcd((N-1)!, N) = 1$，
与N为素数矛盾
得证若N为合数则$(N-1)! \not\equiv -1 (mod\ N)$

1 习题一

1.35

(c) Show that if N is not prime, then $(N-1)! \not\equiv -1\ (mod\ N)$

思路二：讨论N

若$N = ab$，且$a < b$，则$(N-1)! = 1 \times \cdots \times a \times \cdots \times b \times \cdots \times (N-1)$，$(\mathrm{N}-1)! \equiv 0\ (mod\ N)$

若$N = k^2$，当$k > 2$时，$N > 2k$，则$(N-1)! = 1 \times \cdots \times k \times \cdots \times 2k \times \cdots \times (N-1)$，$(\mathrm{N}-1)! \equiv 0\ (mod\ N)$

若$N = k^2$且$k \leq 2$，则$N = 1$或4

$N = 1$时，$(N-1)! \equiv 1 \equiv 0\ (mod\ 1)$

$N = 4$时，$(N-1)! \equiv 3! \equiv 6 \equiv 2 \not\equiv -1\ (mod\ 4)$

1 习题一

1.35

(d) Unlike Fermat's Little theorem, Wilson's theorem is an if-and-only-if condition for primality. Why can't we immediately base a primality test on this rule?

用费马小定理检验素数时需计算 a^{N-1}，可用快速幂算法，复杂度为$\mathrm{O}(\log N)$；用Wilson定理则需要计算$(N-1)!$，复杂度为$\mathrm{O}(N)$

习题三

2 习题三

3.7

(a) Give a linear-time algorithm to determine whether an undirected graph is bipartite

1、从当前尚未被染过色的节点中随机取一个点v，设v的颜色$color[v] = 0$

2、从v开始遍历v所在的连通块（BFS/DFS）

3、设当前遍历到点u，考察u的邻居t
若t未被染色，则t的颜色$color[t] = color[u]\ xor\ 1$，之后会从$t$开始继续遍历
若t已被染过色，则判断t的颜色是否与u相同
若不同，则继续遍历
若相同，则发生染色冲突，说明原图不是二分图

4、重复上述遍历过程，直到遍历完整张图
若遍历过程中未发生染色冲突，则说明该图为二分图

2 习题三

3.7

(b) Prove the following formulation: an undirected graph is bipartite if and only if it contains no cycles of odd length

由(a)可知，当(a)中所进行的遍历过程中不出现染色冲突，则说明该图是二分图

证明出现染色冲突⇔ 图中存在奇数长的环

1、若从点u遍历到邻居t时，发生染色冲突，即$color[u] = color[v]$，证明u,t在一个奇数长的环中

考察u,t在遍历树中的最近公共祖先v_0，则路径$v_0 - u$和$v_0 - t$的长度是同奇偶的，因此环$v_0 - u - t - v_0$长度必为奇数

2、证明若存在奇数环，则该环上必发生染色冲突

设该奇数环为$v_0 - v_1 - \cdots - v_{2k} - v_0$，考察遍历路径$v_0 - v_1 - \cdots - v_{2k}$并染色，则下标为偶数的点$v_0, v_2, \cdots, v_{2k}$两两同色，下标为奇数的点$v_1, v_3, \cdots, v_{2k-1}$两两同色，因此该奇数环上的边$v_{2k} - v_0$发生染色冲突

2 习题三

3.7

(c) At most how many colors are needed to color in an undirected graph with exactly one odd-length cycle?

由(b)可知，只用2种颜色是不行的，所以需要的颜色数≥ 2，下证3种颜色可行

思路一：正面构造

1、对图中不包含奇数长的环的连通块，可以用3种颜色中任意2种染色

2、考察图中包含该奇数长的环的连通块
首先用3种颜色对奇数环染色
从已染好色的奇数环开始对整个联通块染色
讨论多种情况

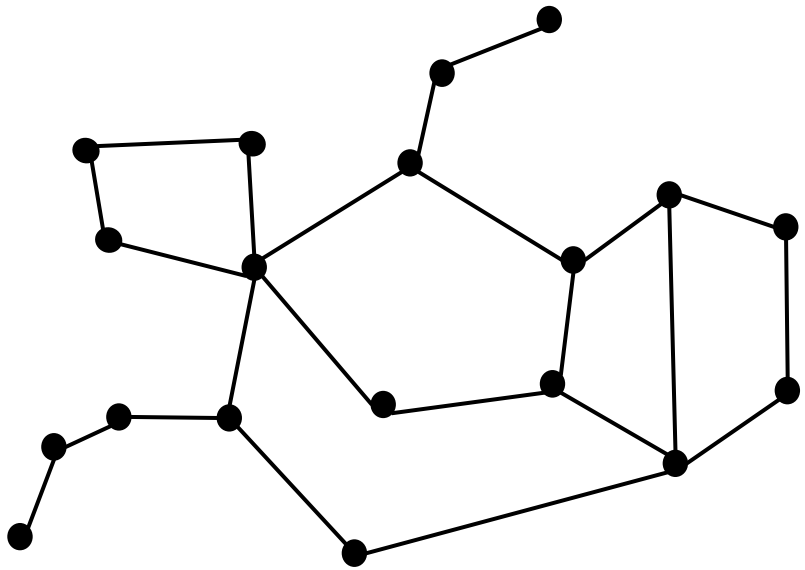
2 习题三

3.7

(c) At most how many colors are needed to color in an undirected graph with exactly one odd-length cycle?

由(b)可知，只用2种颜色是不行的，所以需要的颜色数≥ 2，下证3种颜色可行

思路一：正面构造



2 习题三

3.7

(c) At most how many colors are needed to color in an undirected graph with exactly one odd-length cycle?

由(b)可知，只用2种颜色是不行的，所以需要的颜色数≥ 2，下证3种颜色可行

思路一：正面构造

1、对图中不包含奇数长的环的连通块，可以用3种颜色中任意2种染色

2、考察图中包含该奇数长的环的连通块
首先用3种颜色对奇数环染色
从已染好色的奇数环开始对整个联通块染色
讨论多种情况

不建议从正面构造

2 习题三

3.7

(c) At most how many colors are needed to color in an undirected graph with exactly one odd-length cycle?

由(b)可知，只用2种颜色是不行的，所以需要的颜色数≥ 2，下证3种颜色可行

思路二：缩点

1、在奇数环上选出两相邻节点v_0, v_1合成新节点v'

2、得到的新图不存在奇数环，由(b)可知新图可以用2种颜色染色

3、将v'重新拆开，恢复到原图，原图中v_0, v_1与染色与v'相同，其余点染色不变

4、则此时原图中只有这两个边$v_0 - v_1$存在染色冲突
将v_0或v_1染色改为第三种颜色，则得到对该图的合法染色

2 习题三

3.11

Design a linear-time algorithm which, given an undirected graph G and a particular edge e in it,determines whether G has a cycle containing e.

设边e为$e(u,v)$，将e从图中去掉后，从u开始遍历，若能遍历到v则说明原图中存在包含e的环；反之，则不存在包含e的环

2 习题三

3.28

(a) Are there other satisfying truth assignments of this 2SAT formula? If so, find them all.

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge (\overline{x_3} \vee x_4) \wedge (\overline{x_1} \vee x_4)$$

由$(x_1 \vee \overline{x_2})$和$(x_1 \vee x_2)$，可知$x_1 = 1$

由$(\overline{x_1} \vee \overline{x_3})$和$(\overline{x_1} \vee x_4)$，可知$x_3 = 0$，$x_4 = 1$，且满足$(\overline{x_3} \vee x_4) = 1$

则x_2可任取

$x_1 = 1$，$x_2 = 0$，$x_3 = 0$，$x_4 = 1$

$x_1 = 1$，$x_2 = 1$，$x_3 = 0$，$x_4 = 1$

2 习题三

3.28

(b) Give an instance of 2SAT with four variables, and with no satisfying assignment

$$(x_1 \vee \overline{x_2}) \wedge (x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2) \wedge (x_3 \vee x_4)$$

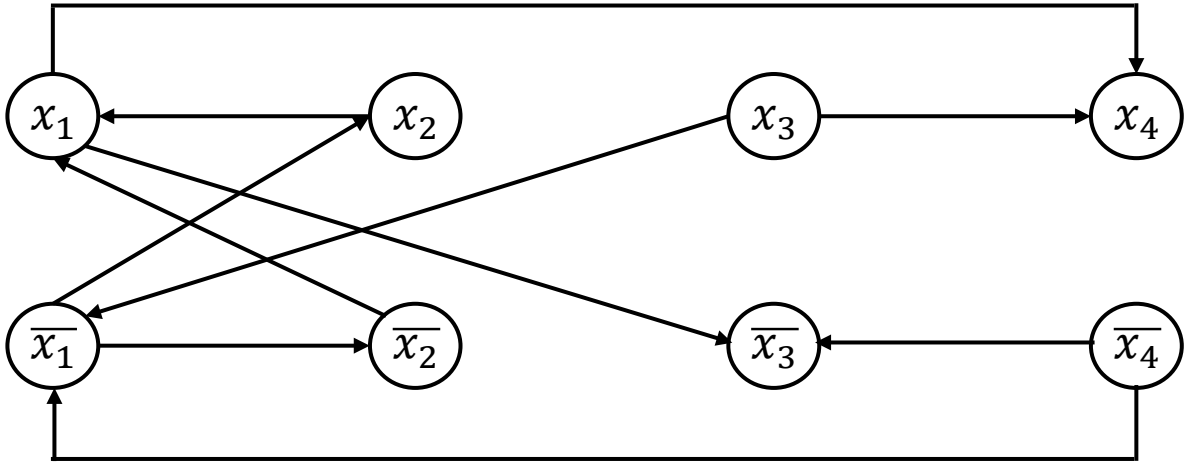
2 习题三

3.28

(c) Carry out this construction for the instance of 2SAT given above, and for the instance you constructed in (b)

Clause $(\alpha \vee \beta)$ is equivalent to either of the implications $\bar{\alpha} \Rightarrow \beta$ or $\bar{\beta} \Rightarrow \alpha$

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge (\overline{x_3} \vee x_4) \wedge (\overline{x_1} \vee x_4)$$



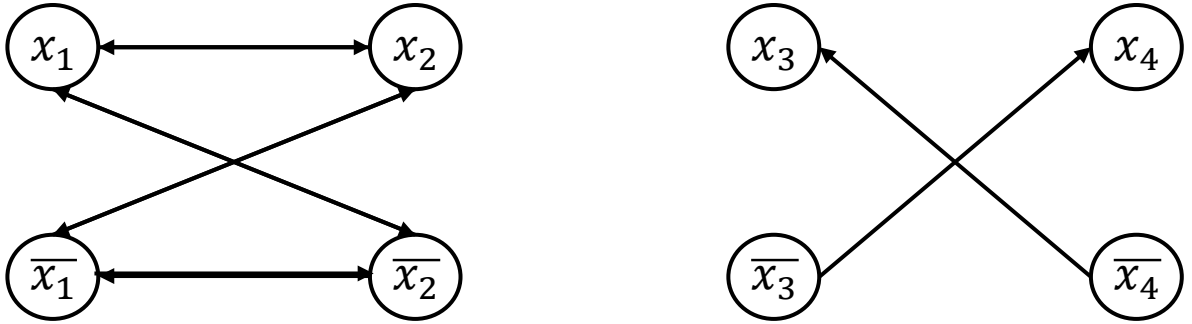
2 习题三

3.28

(c) Carry out this construction for the instance of 2SAT given above, and for the instance you constructed in (b)

Clause $(\alpha \vee \beta)$ is equivalent to either of the implications $\bar{\alpha} \Rightarrow \beta$ or $\bar{\beta} \Rightarrow \alpha$

$$(x_1 \vee \overline{x_2}) \wedge (x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2) \wedge (x_3 \vee x_4)$$



2 习题三

3.28

(d) Show that if G_I has a strongly connected component containing both $\bar{x}$ and x for some variable x, then I has no satisfying assignment

Clause $(\alpha \vee \beta)$ is equivalent to either of the implications $\bar{\alpha} \Rightarrow \beta$ or $\bar{\beta} \Rightarrow \alpha$

存在环$x \Rightarrow x_1 \Rightarrow \cdots \Rightarrow x_k \Rightarrow \bar{x} \Rightarrow x_{k+1} \Rightarrow \cdots \Rightarrow x_{k+n} \Rightarrow x$
假设存在一个真值指派使I为真，讨论这个真值指派中x的取值
若$x = 1$，考察路径$x \Rightarrow x_1 \Rightarrow \cdots \Rightarrow x_k \Rightarrow \bar{x}$，该路径表示$I$包含合取范式
$(\bar{x} \vee x_1) \wedge (\overline{x_1} \vee x_2) \wedge \cdots \wedge (\overline{x_k} \vee \bar{x})$，$x = 1 \Rightarrow x_1 = 1 \Rightarrow x_2 = 1 \Rightarrow \cdots \Rightarrow x_k = 1$，
则$\overline{x_k} \vee \bar{x} = 0$，不满足
若$x = 0$，考察路径$\bar{x} \Rightarrow x_{k+1} \Rightarrow \cdots \Rightarrow x_{k+n} \Rightarrow x$，该路径表示$I$包含合取范式
$(x \vee x_{k+1}) \wedge (\overline{x_{k+1}} \vee x_{k+2}) \wedge \cdots \wedge \left(\overline{x_{k+j}} \vee x\right)$，$x = 0 \Rightarrow x_{k+1} = 1 \Rightarrow x_{k+2} = 1 \Rightarrow$
$\cdots \Rightarrow x_{k+j} = 1$，则$\overline{x_k} \vee x = 0$，不满足
则I的真值指派不存在，I不可满足

2 习题三

3.28

(e) Now show the converse of (d): namely, that if none of G_I 's strongly connected components contain both a literal and its negation, then the instance I must be satisfiable.

证明存在从x到$\bar{x}$的强连通分量是I不可满足的充要条件

存在从x到$\bar{x}$的强连通分量$\Rightarrow I$不可满足，见(d)

I不可满足$\Rightarrow$存在从x到$\bar{x}$的强连通分量

$(\alpha \vee \beta)$等价于蕴含关系 $\bar{\alpha} \to \beta \vee \bar{\beta} \to \alpha$

G_I表示了I的每一个二元析取语句所转化出的等价蕴含关系

由蕴含关系的可传递性，I中的蕴含关系都与G_I中的路径一一对应

I不可满足 $\Rightarrow I$中存在变量x在I中有蕴含关系$x \to \bar{x}$且$\bar{x} \to x$

即G_I中存在从x到$\bar{x}$和从$\bar{x}$到x的路径，得到一个从x到$\bar{x}$的强连通分量

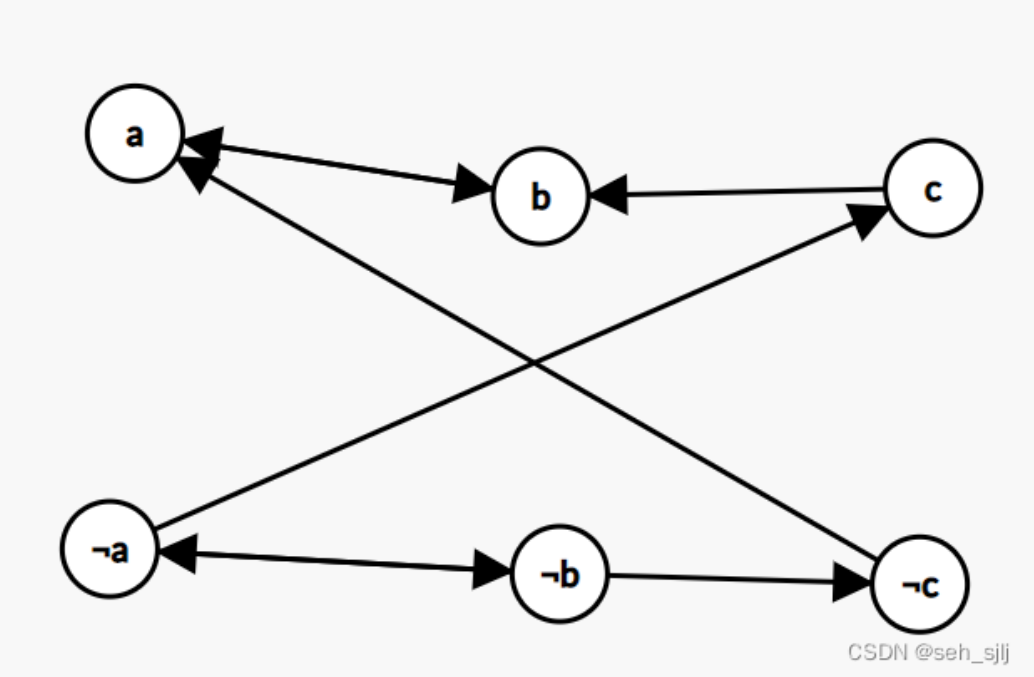
2 习题三

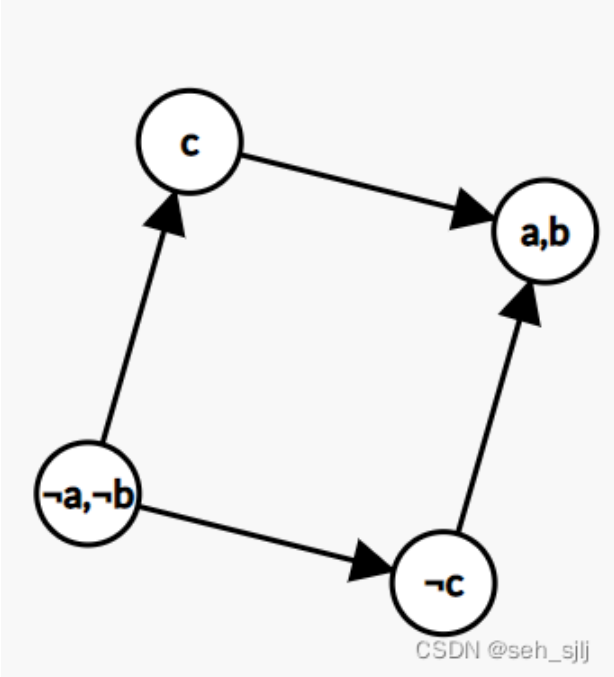
3.28

(f) Conclude that there is a linear-time algorithm for solving 2SAT.

由于不存在从x到$\bar{x}$的强连通分量，所以整张图不是一个强连通分量

将每个强连通分量缩为一个点，由拓扑排序可知存在一个强连通分量对应的点的拓扑序最高，设该连通分量为V，则可知不存在从V中点指向V外点的边





2 习题三

3.28

(f) Conclude that there is a linear-time algorithm for solving 2SAT.

显然存在另一个强连通分量$\bar{V}$，$\bar{V}$中所有点均为V中点的反

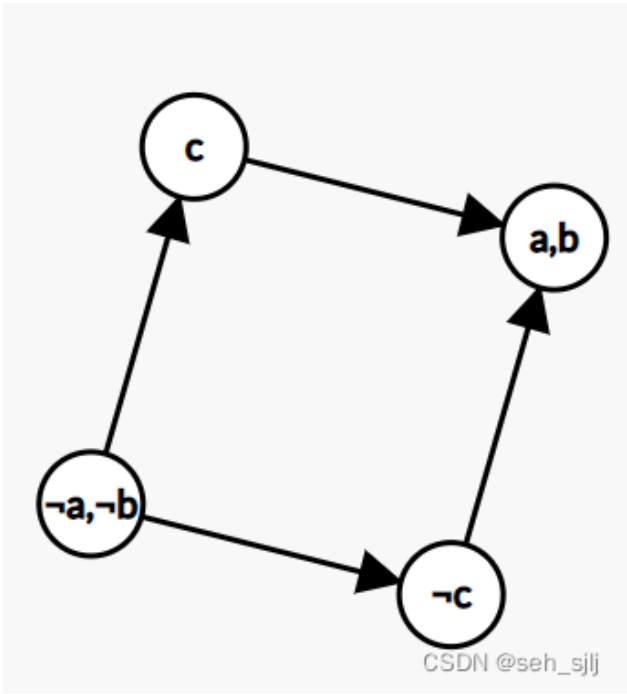
（引理1：若存在x到y的路径，则存在$\bar{y}$到$\bar{x}$的路径

证明：设x到y的路径为$x \to v_0 \to \cdots \to v_k \to y$，则存在路径$\bar{x} \leftarrow \overline{v_0} \leftarrow \cdots \leftarrow \overline{v_k} \leftarrow \bar{y}$

引理2：若x和y属于同一个强连通分量，则$\bar{x}$和$\bar{y}$属于同一个强连通分量

证明：x和y属于同一个强连通分量，则存在x到y和y到x的路径；由引理1可知，存在$\bar{y}$到$\bar{x}$和$\bar{x}$到$\bar{y}$的路径，故$\bar{x}$和$\bar{y}$属于同一个强连通分量）

则可知不存在从$\bar{V}$外点指向$\bar{V}$中点的边



2 习题三

3.28

(f) Conclude that there is a linear-time algorithm for solving 2SAT.

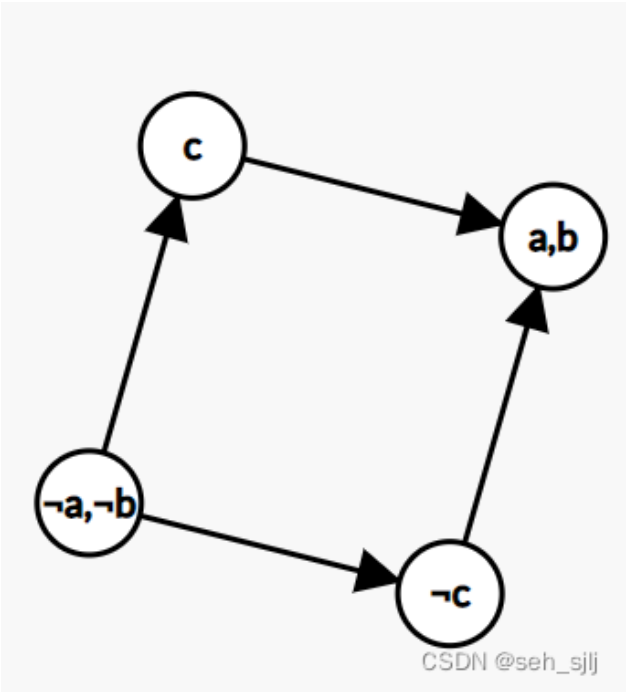
对V中所有点均赋值为1，$\bar{V}$中所有点赋值为0，对V和$\bar{V}$内部的连边是满足的

（证明：对V中的路径$x_1 \to x_2 \to \cdots \to x_k$（$x_1, x_2, \cdots, x_k$对应为形同$a$或$\bar{b}$的变量），有$x_1 = 1 \Rightarrow x_2 = 1 \Rightarrow \cdots \Rightarrow x_k = 1$，即$x_1 = x_2 = \cdots = x_k = 1$是可满足的；$\bar{V}$中对应路径为$\overline{x_1} \leftarrow \overline{x_2} \leftarrow \cdots \leftarrow \overline{x_k}$，则$\overline{x_1} = \overline{x_2} = \cdots = \overline{x_k} = 0$同样可满足）

对任一条从V外点指向V内点的边$e: u \to v$，由$v = 1$可知$u \to v$恒为真

对任一条从$\bar{V}$内点指向$\bar{V}$外点的边$e: \bar{v} \to u$，由$\bar{v} = 0$可知$\bar{v} \to u$恒为真

因此可以将强连通分量V和$\bar{V}$从G_I中删除，然后继续按拓扑序从高到低为每一个强连通分量赋值



2 习题三

3.28

(f) Conclude that there is a linear-time algorithm for solving 2SAT.

Tarjan算法求强连通分量+缩点
拓扑排序+赋值

2 习题三

4.11

Give an algorithm that takes as input a directed graph with positive edge lengths, and returns the length of the shortest cycle in the graph (if the graph is acyclic, it should say so). Your algorithm should take time at most $O(|V|^3)$.

思路一:

$|V|$次Dijkstra算法，对每一个节点求出其到其他节点的最短路大小

对每一条边$e: u \to v$，包含点u, v的最短环长度为$dist(u, v) + dist(v, u)$

取所有最短环长度中的最小值，即得到图中最短环长度，不存在环时得到∞

思路二:

将$|V|$次Dijkstra算法更改为一次Floyd算法得到任意两点的最短路大小

2 习题三

4.11

Give an algorithm that takes as input a directed graph with positive edge lengths, and returns the length of the shortest cycle in the graph (if the graph is acyclic, it should say so). Your algorithm should take time at most $O(|V|^3)$.

思路三：Floyd求最短环，松弛到节点k时，用节点序号最大为k的最小环更新ans

```
int ans=inf;
for(int k=1; k<=n; k++){
   for(int i=1; i<k; i++)
      for(int j=i+1; j<k; j++)
         ans=min(ans,f[i][j]+graph[j][k]+graph[k][i]);
   for(int i=1; i<=n; i++)
      for(int j=1; j<=n; j++)
         if(f[i][k]+f[k][j]<=f[i][j])
            f[i][j]=f[i][k]+f[k][j];
   }
```

2 习题三

4.12

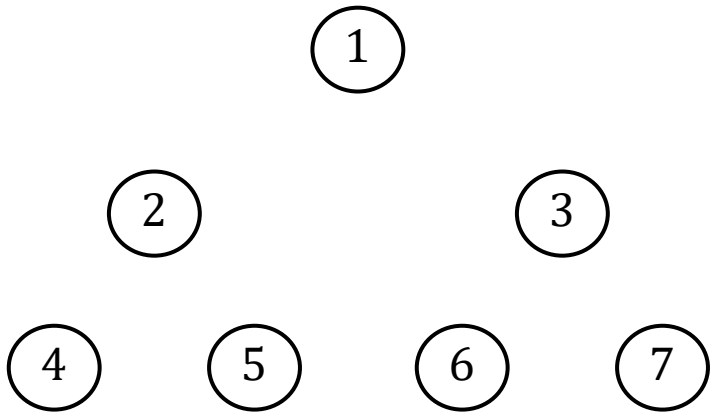
Give an undirected graph $G = (V, E)$ whose edge lengths > 0 and an edge $e \in E$. Compute the length of the shortest cycle containing edge e in $O(|V|^2)$.

设$e = (u, v)$，从G中删除e，然后求出u到v的最短路$dist(u, v)$，再加上e的长度l_e得到$dist(u, v) + l_e$，即为G中包含e的最短环长度。若$dist(u, v) + l_e = \infty$则说明$G$中不存在包含$e$的环

2 习题三

4.16

(a) Consider the node at position j of the array. Show that its parent is at position $\lfloor j/2 \rfloor$ and its children are at $2j$ and $2j + 1$ (if these numbers are $\leq n$)



设下标为j的节点为第m层第n个节点，j,m,n均从1开始，则前$m-1$层共有$2^{m-1}-1$个节点，$j=2^{m-1}+n-1$

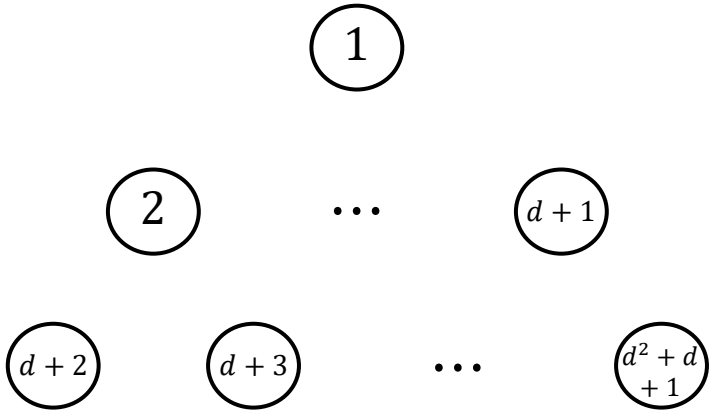
则其父节点为第$m-1$层第$\left\lfloor\frac{n+1}{2}\right\rfloor$个节点，则父节点下标为$2^{m-2}+\left\lfloor\frac{n+1}{2}\right\rfloor-1=2^{m-2}+\left\lfloor\frac{n-1}{2}\right\rfloor=\left\lfloor\frac{2^{m-1}+n-1}{2}\right\rfloor=\left\lfloor\frac{j}{2}\right\rfloor$

其子节点下标k则满足$\left\lfloor\frac{k}{2}\right\rfloor=j$，则$k=2j$或$2j+1$

2 习题三

4.16

(b) What the corresponding indices when a complete d-ary tree is stored in an array?



设下标为j的节点为第m层第n个节点，j,m,n均从1开始，则前$m-1$层共有$\frac{d^{m-1}-1}{d-1}$个节点，$j=\frac{d^{m-1}-1}{d-1}+n$

则其父节点为第$m-1$层第$\left\lfloor\frac{n+d-1}{d}\right\rfloor$个节点，则父节点下标为$\frac{d^{m-2}-1}{d-1}+\left\lfloor\frac{n+d-1}{d}\right\rfloor=\left\lfloor\frac{d^{m-1}-d}{(d-1)d}+\frac{n+d-1}{d}\right\rfloor=\left\lfloor\frac{(j-n)(d-1)+1-d}{(d-1)d}+\frac{n+d-1}{d}\right\rfloor=\left\lfloor\frac{j-n-1+n+d-1}{d}\right\rfloor=\left\lfloor\frac{j+d-2}{d}\right\rfloor$

其子节点下标k则满足$\left\lfloor\frac{k+d-2}{d}\right\rfloor=j$，则$k=dj-d+2,dj-d+1,\cdots,dj+1$

2 习题三

4.16 $|h|$, which returns the number of elements currently in the array;

```
function makeheap(S)
h = empty array of size |S|
for x ∈ S:
    h(|h| + 1) = x
for i = |S| downto 1:
    siftdown(h, h(i), i)
return h
```

将S中的元素以对应的key值为基准构造一个小根堆，根节点下标为1

```
procedure siftdown(h, x, i)
(place element x in position i of h, and let it sift down)
c = minchild(h, i)
while c ≠ 0 and key(h(c)) < key(x):
    h(i) = h(c);  i = c;  c = minchild(h, i)
h(i) = x
```

若当前节点的key值比其$minchild$节点小，则将该节点下移

```
function minchild(h, i)
(return the index of the smallest child of h(i))
if 2i > |h|:
    return 0 (no children)
else:
    return arg min{key(h(j)) : 2i ≤ j ≤ min{|h|, 2i + 1}}
```

得到2个子节点中key值更小的一个

2 习题三

4.16 $|h|$, which returns the number of elements currently in the array;

```
function deletemin(h)
if |h| = 0:
    return null
else:
    x = h(1)
    siftdown(h, h(|h|), 1)
    return x
```

删除当前对应的key值最小的元素，即堆顶元素，并维护堆

```
procedure siftdown(h, x, i)
(place element x in position i of h, and let it sift down)
c = minchild(h, i)
while c ≠ 0 and key(h(c)) < key(x):
    h(i) = h(c);  i = c;  c = minchild(h, i)
h(i) = x

function minchild(h, i)
(return the index of the smallest child of h(i))
if 2i > |h|:
    return 0 (no children)
else:
    return arg min{key(h(j)) : 2i ≤ j ≤ min{|h|, 2i + 1}}
```

2 习题三

4.16 $|h|$, which returns the number of elements currently in the array;
h^{-1}, which returns the position of an element within the array

```
procedure insert(h, x)
bubbleup(h, x, |h| + 1)
```

向堆中加入新元素，将新元素添加到堆底再上移到合适位置

```
procedure decreasekey(h, x)
bubbleup(h, x, h^{-1}(x))
```

减小了某个元素对应的key值后，需判断该元素是否需要上移

```
procedure bubbleup(h, x, i)
(place element x in position i of h, and let it bubble up)
p = ⌈i/2⌉
while i ≠ 1 and key(h(p)) > key(x):
    h(i) = h(p);  i = p;  p = ⌈i/2⌉
h(i) = x
```

若当前节点的key值比其父节点小，则将该节点上移

2 习题三

4.16

(c) Show that the *makeheap* procedure takes $O(n)$ time when called on a set of n elements. What is the worst-case input?

```
function makeheap(S)
h = empty array of size |S|
for x ∈ S:
    h(|h| + 1) = x
for i = |S| downto 1:
    siftdown(h, h(i), i)
return h
```

这个*makeheap*的思路是：将队列先全部扔进堆里，然后从叶子节点开始维护堆，保证所有节点元素的*key*值都比其子节点元素小，否则就将节点元素与其*minchild*节点元素交换，不断下移直到满足小根堆

所以考察第i个节点，最多移到堆底，交换次数为$\log n - \log i$，复杂度为$O(\log \frac{n}{i})$；总的算法复杂度即为

$$O\left(\sum_{i=1}^{n} \log \frac{n}{i}\right) = O\left(\log \frac{n^n}{n!}\right)$$

由斯特林公式得$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$，则$O\left(\log \frac{n^n}{n!}\right) =$

$$O\left(\log \frac{n^n}{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n}\right) = O\left(\log \frac{e^n}{\sqrt{2\pi n}}\right) = O(n)$$

构造最坏解——保证每个子树的根都是这颗子树里最大的，按倒序$n, n-1, \cdots, 2, 1$输入

2 习题三

4.16

(d) What needs to be changed to adapt this pseudocode to d-ary heaps

从当前节点下标计算父、子节点下标的过程需要修改

```
procedure bubbleup(h, x, i)
(place element x in position i of h, and let it bubble up)
p = ⌈i/2⌉
while i ≠ 1 and key(h(p)) > key(x):
    h(i) = h(p);  i = p;  p = ⌈i/2⌉
h(i) = x

function minchild(h, i)
(return the index of the smallest child of h(i))
if 2i > |h|:
    return 0 (no children)
else:
    return arg min{key(h(j)) : 2i ≤ j ≤ min{|h|, 2i + 1}}
```

$$p = \left\lfloor \frac{i + d - 2}{d} \right\rfloor$$

$$di - d + 2 > |h|$$

$$di - d + 2 \le j \le \min\{|h|, di + 1\}$$

习题五

3 习题五

7.6

Give an example of a linear program in two variables whose feasible region is infinite, but such that there is an optimum solution of bounded cost.

Objective function $\min x_1 - x_2$

Constraints $x_1, x_2 \geq 0$

3 习题五

7.7

(a) Find necessary and sufficient conditions on the reals a and b under which the linear program max x + y ax + by ≤ 1 x, y ≥ 0 is infeasible

发现$x=0, y=0$时恒满足约束条件，所以无论a, b取何值始终是feasible

3 习题五

7.7

(b) Find necessary and sufficient conditions on the reals a and b under which the linear program max x + y ax + by ≤ 1 x, y ≥ 0 is unbounded.

$a \leq 0$时取$x \to +\infty, y = 0$恒满足约束条件

$b \leq 0$时取$x = 0, y \to +\infty$恒满足约束条件

当$a > 0$且$b > 0$时，有$x \leq \frac{1}{a}, y \leq \frac{1}{b}$，$x + y \leq \frac{1}{a} + \frac{1}{b}$有界

所以$a \leq 0$或$b \leq 0$时该线性规划无界

3 习题五

7.7

(c) Find necessary and sufficient conditions on the reals a and b under which the linear program max x + y ax + by ≤ 1 x, y ≥ 0 has a unique optimal solution.

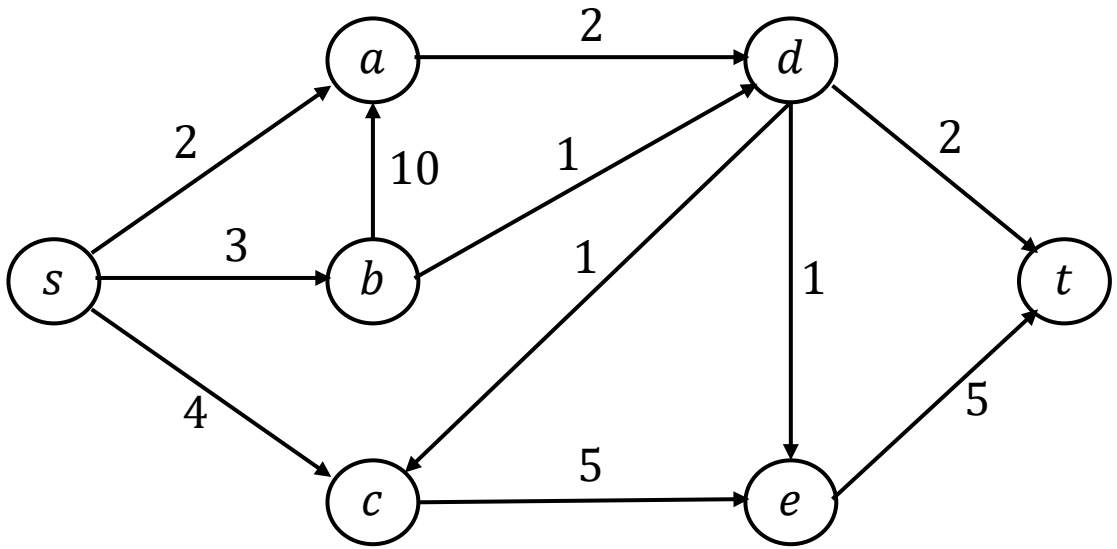
当$a > 0$且$b > 0$时，该线性规划有最优解
当$\mathrm{a} = \mathrm{b}$时，该线性规划的最优解有无数个
所以当$a > 0, b > 0$且$a \neq b$时该线性规划有唯一最优解

3 习题五

7.21

An edge of a flow network is called critical if decreasing the capacity of this edge results in a decrease in the maximum flow. Give an efficient algorithm that finds a critical edge in a network.

首先求最大流，再构造出残量图。显然，临界边的容量肯定被占满，所以在残量图中，临界边的顶点对之间只存在反向边。

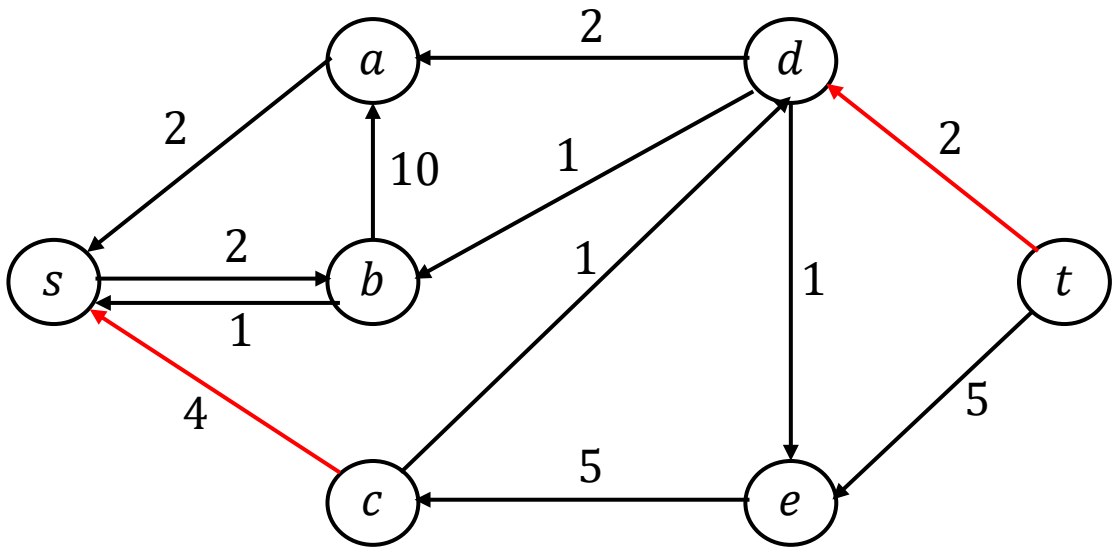


3 习题五

7.21

An edge of a flow network is called critical if decreasing the capacity of this edge results in a decrease in the maximum flow. Give an efficient algorithm that finds a critical edge in a network.

首先求最大流，再构造出残量图。显然，临界边的容量肯定被占满，所以在残量图中，临界边的顶点对之间只存在反向边。

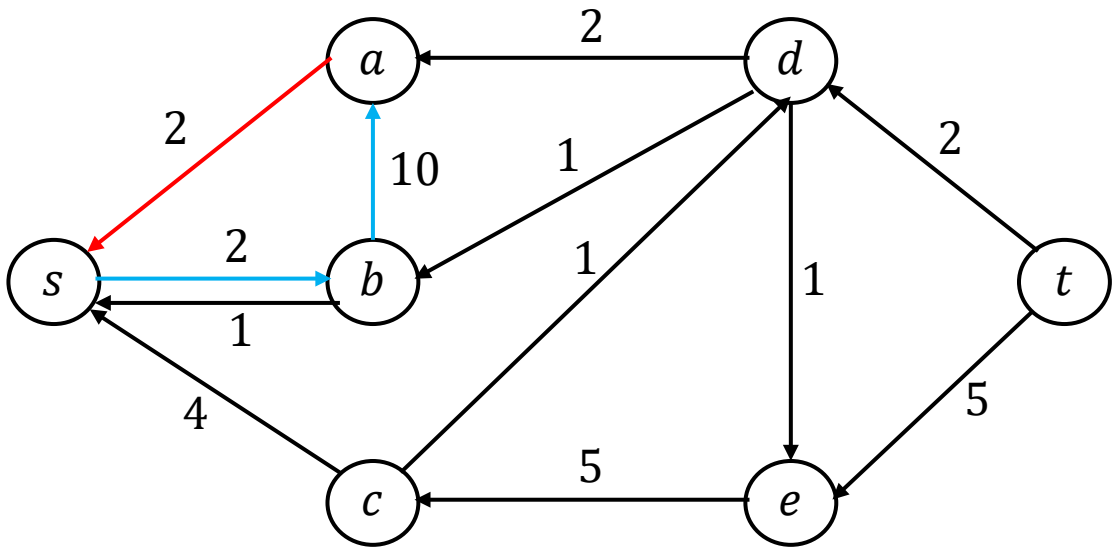


3 习题五

7.21

An edge of a flow network is called critical if decreasing the capacity of this edge results in a decrease in the maximum flow. Give an efficient algorithm that finds a critical edge in a network.

但在残量图中只存在反向边的顶点对不一定都是临界边的顶点，可能某条在最大流中流满的边，其容量下降后，最大流中缺少的流量可以从其他边的流量得到补充

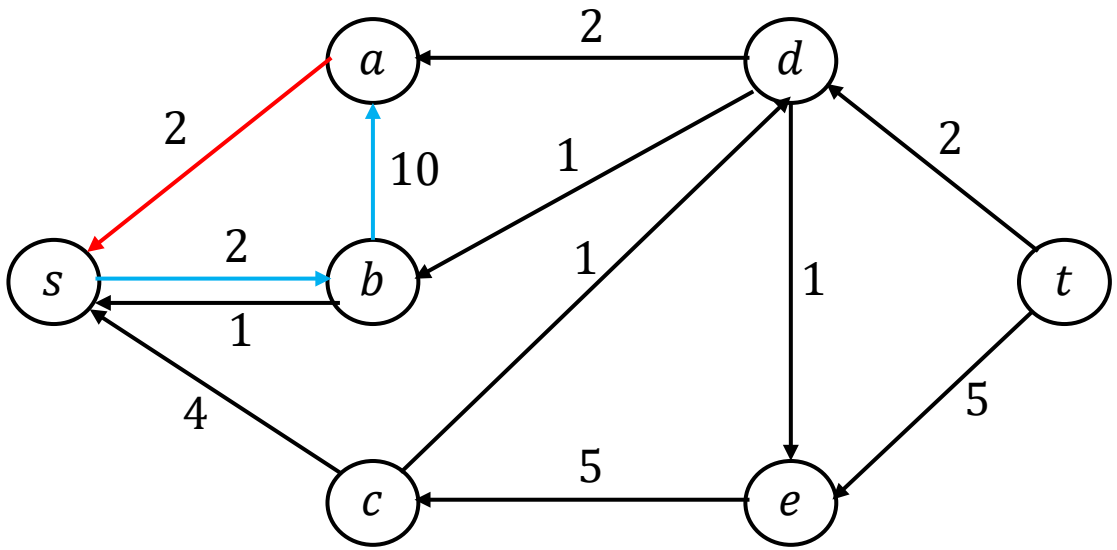


3 习题五

7.21

An edge of a flow network is called critical if decreasing the capacity of this edge results in a decrease in the maximum flow. Give an efficient algorithm that finds a critical edge in a network.

在残量图中以S为源点进行DFS，则所经过的反向边均不是临界边（参考答案）

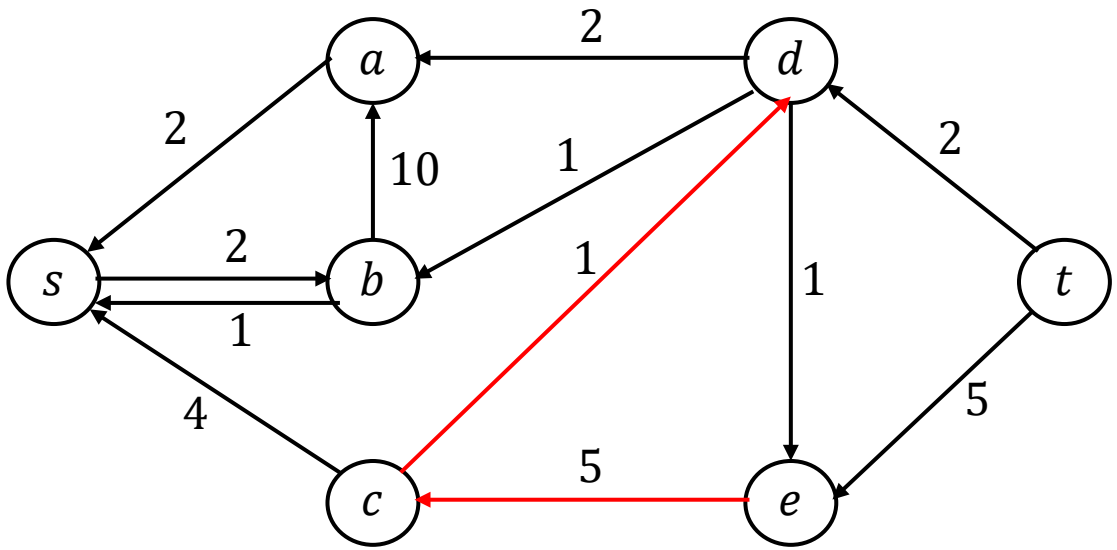


3 习题五

7.21

An edge of a flow network is called critical if decreasing the capacity of this edge results in a decrease in the maximum flow. Give an efficient algorithm that finds a critical edge in a network.

在残量图中以S为源点进行DFS，则所经过的反向边均不是临界边——不正确，因为没去除掉所有的非临界边的反向边



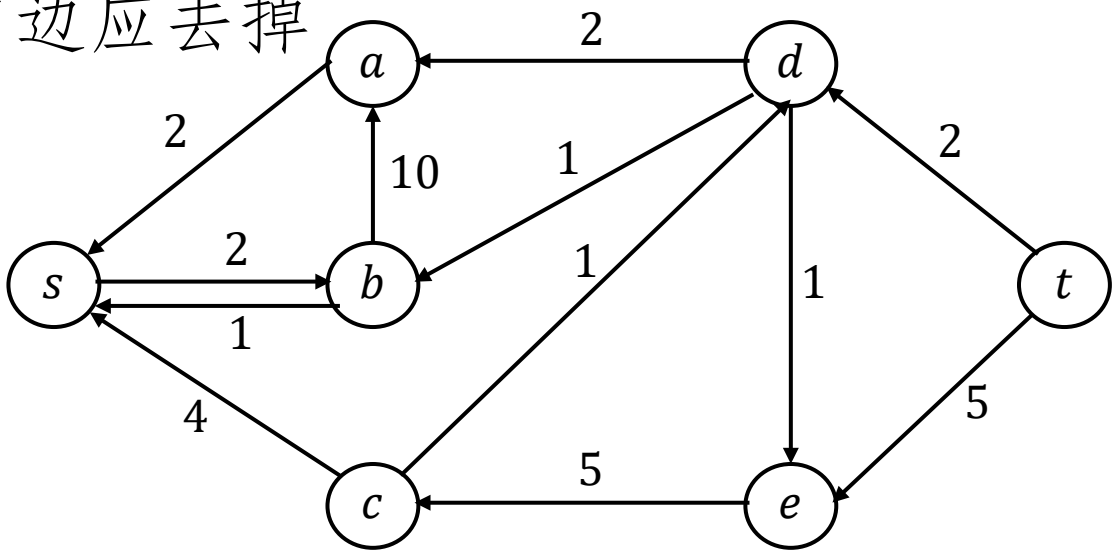
3 习题五

7.21

An edge of a flow network is called critical if decreasing the capacity of this edge results in a decrease in the maximum flow. Give an efficient algorithm that finds a critical edge in a network.

（核心）容量下降后，最大流中缺少的流量可以从其他边的流量得到补充

因此对每条反向边，从汇点DFS，若能DFS到源点，就说明该反向边应去掉



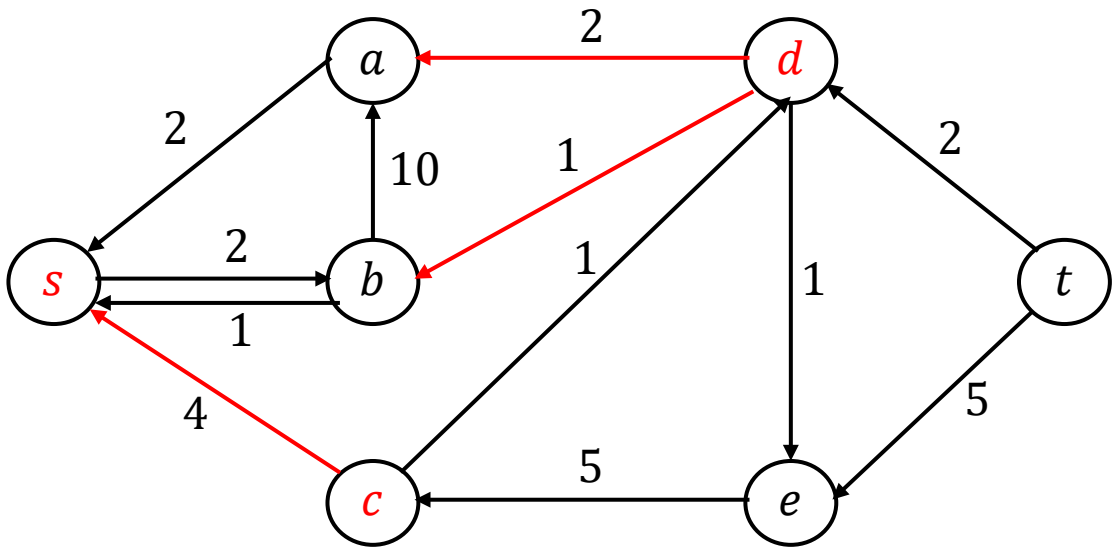
3 习题五

7.21

An edge of a flow network is called critical if decreasing the capacity of this edge results in a decrease in the maximum flow. Give an efficient algorithm that finds a critical edge in a network.

（更好的方法）源点的DFS因最小割而无法继续继续

$e:a\to d, e:b\to d$和$e:s\to c$均为临界边，为继续求出不合格的反向边，需要以为c,d为次级源点继续DFS

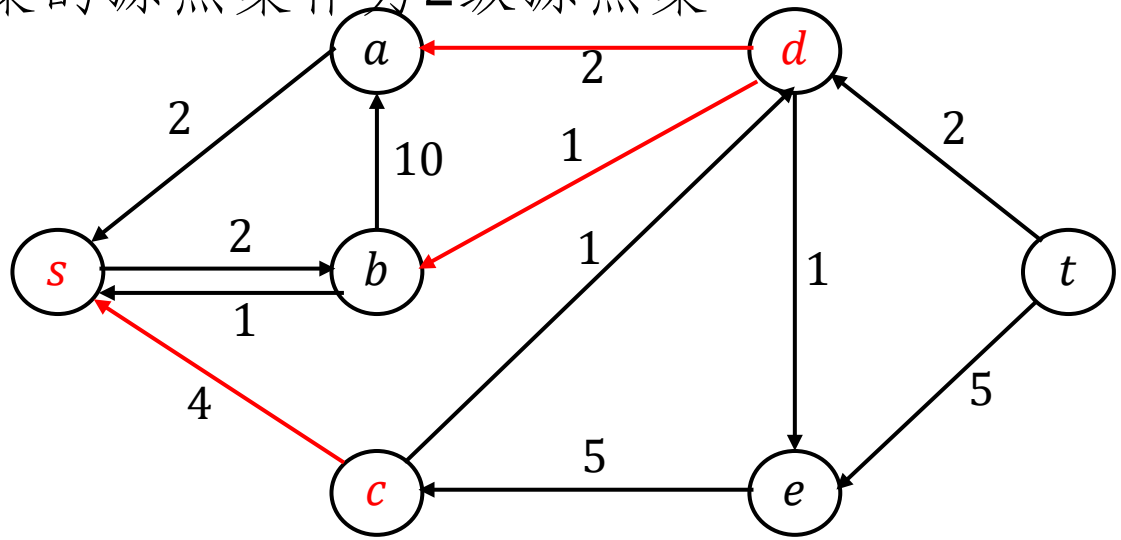


3 习题五

7.21

An edge of a flow network is called critical if decreasing the capacity of this edge results in a decrease in the maximum flow. Give an efficient algorithm that finds a critical edge in a network.

（更好的方法）从源点开始DFS，遍历到的点集和未遍历到的点集形成图的一个割，割边集中所有的反向边均对应一条临界边；去掉遍历到的子图和割边集，得到了原图的一个未被遍历的子图，割边集的源点集作为2级源点集

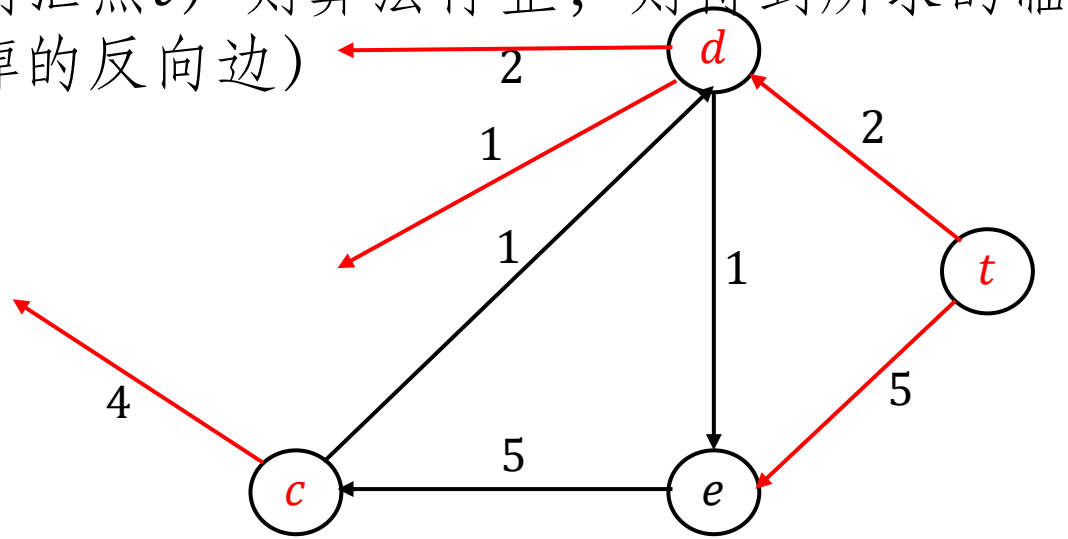


3 习题五

7.21

An edge of a flow network is called critical if decreasing the capacity of this edge results in a decrease in the maximum flow. Give an efficient algorithm that finds a critical edge in a network.

（更好的方法）从i级源点继续DFS，若遍历到的点集和未遍历到的点集同样形成图的一个割，则重复上述操作，得到割边集、新的未被遍历的子图，作为 $i+1$ 级源点集；若遍历完整张（子）图（即遍历到汇点t）则算法停止，则得到所求的临界边集（或所有应该去掉的反向边）

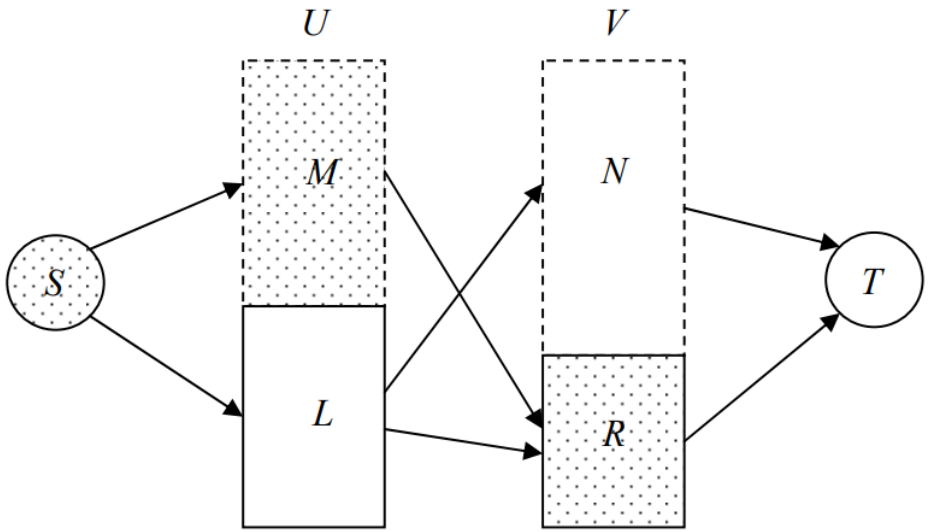


3 习题五

7.23

Show that the problem of finding the minimum vertex cover in a bipartite graph reduces to maximum flow. (Hint: Can you relate this problem to the minimum cut in an appropriate network?)

答案的缺陷：找到的一组割$(S \cup M \cup R, L \cup N \cup T)$=最小点覆盖，但未证明该割即为最小割，因此还需要证明对任意一种割，其割边数≤割$(S \cup M \cup R, L \cup N \cup T)$的割边数（证明思路：讨论不属于$S \to L$和$R \to T$的割边，则添加新的割边会导致$S \to L$和$R \to T$中的割边减少，将割边数转移到节点数，利用最小点覆盖说明总割边数不减性）



3 习题五

7.23

Show that the problem of finding the minimum vertex cover in a bipartite graph reduces to maximum flow. (Hint: Can you relate this problem to the minimum cut in an appropriate network?)

思路二：最小点覆盖 ⇒最小割

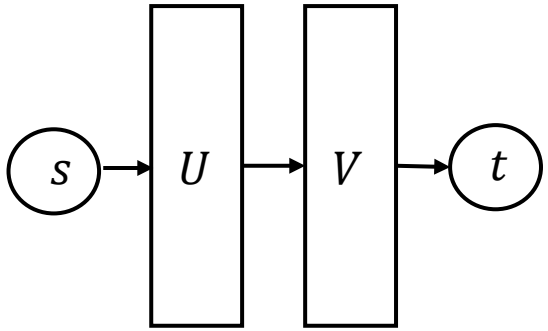
设二分图$G=(U\cup V,E)$，则考察如下的线性规划：

Objective function $\min \sum_{u\in U} a_u + \sum_{v\in V} a_v$

Constraints $u\in U, v\in V$

$\forall i\in U\cup V, a_i\in\{0,1\}$

$a_u + a_v \geq 1$，若$e: u\to v\in E$



可知该线性规划等价于G的最小点覆盖

且如右图所示 网络，各边权为1，则该线性规划同样等价于该网络的最小割（证明，若$e: u\to v\in E$，对流$s\to u\to v\to t$，则$s\to u$，$u\to v$和$v\to t$三边至少取走一边，对应点u,v至少取走一个，a_u和a_v至少一个为1，即对应$a_u + a_v \geq 1$）

3 习题五

7.23

Show that the problem of finding the minimum vertex cover in a bipartite graph reduces to maximum flow. (Hint: Can you relate this problem to the minimum cut in an appropriate network?)

思路三：最小点覆盖 ⇒ 最大匹配 ⇒最大流

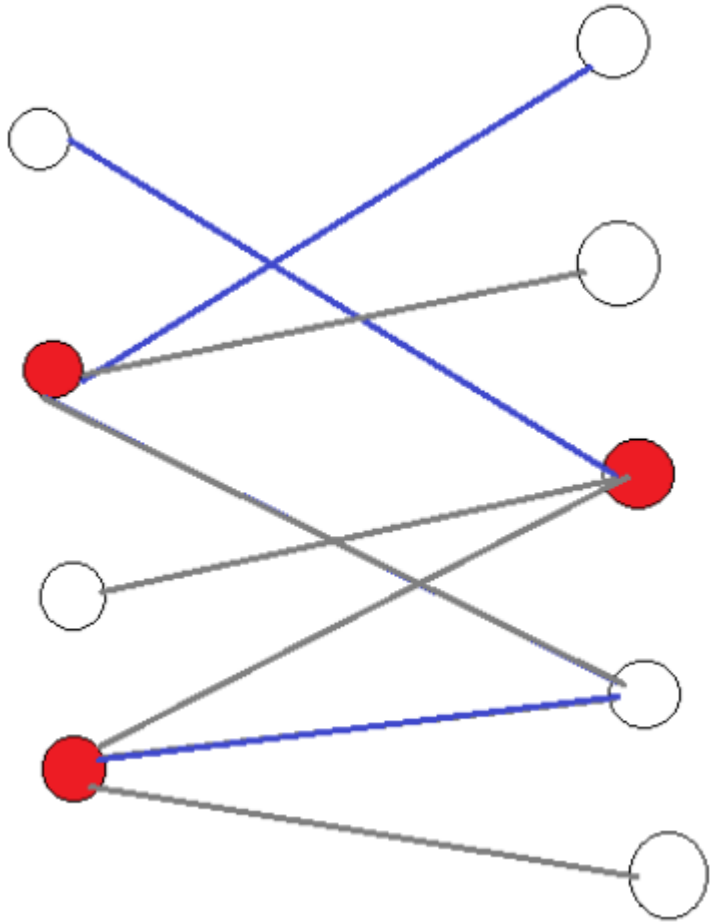
最小点覆盖数≥最大匹配数

从左边非匹配点开始，延非匹配边走向右边的点；右边的点再延匹配边，标记所有走过的点

左边未标记过的点和右边标记过的点集构成一个最小点覆盖（证明：

1、选出来的点集大小=最大匹配

若左边选出的点存在非匹配点，则该点会作为起点被标记；若右边选出的点存在非匹配点，则存在一对非匹配点相连，得到一条新的匹配边，与最大匹配矛盾）



3 习题五

7.23

Show that the problem of finding the minimum vertex cover in a bipartite graph reduces to maximum flow. (Hint: Can you relate this problem to the minimum cut in an appropriate network?)

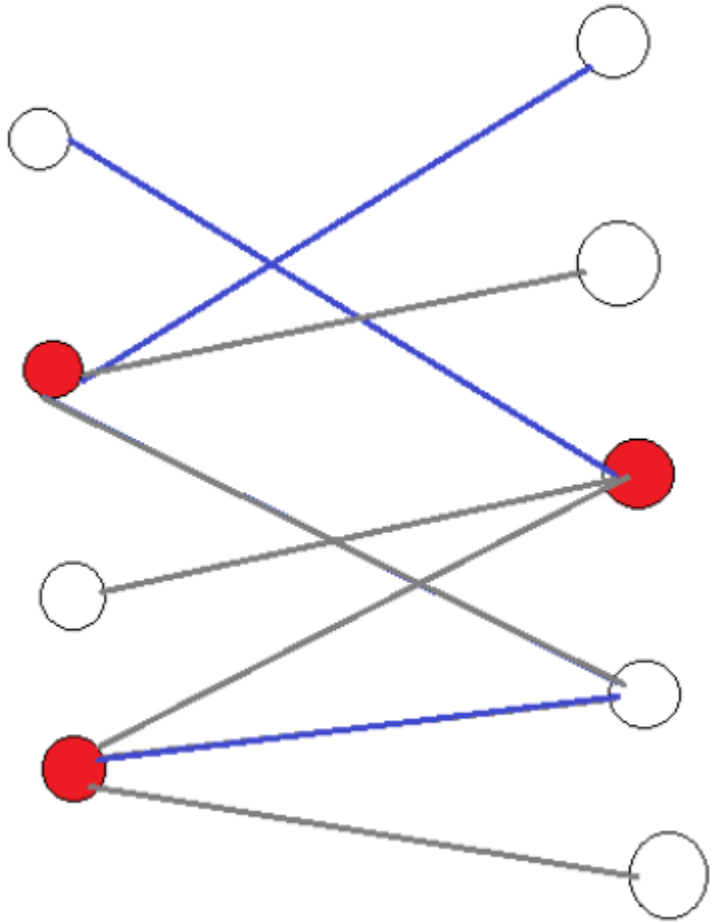
思路三：最小点覆盖 ⇒ 最大匹配 ⇒最大流

左边未标记过的点和右边标记过的点集构成一个最小点覆盖（证明：

2、选出来的点集可以覆盖所有边

因为不可能存在某一条边，其右端点没有标记，而左端点有标记的。

如果这条边不属于匹配边，那么右端点就可以通过这条边得到标记；如果这条边属于匹配边，那么左端点不可能是一条路径的起点，于是其标记只能是从这条边的右端点过来的，右端点就应该有标记。）



3 习题五

7.23

Show that the problem of finding the minimum vertex cover in a bipartite graph reduces to maximum flow. (Hint: Can you relate this problem to the minimum cut in an appropriate network?)

思路三：最小点覆盖 ⇒ 最大匹配 ⇒最大流

左边未标记过的点和右边标记过的点集构成一个最小点覆盖（证明：

1、选出来的点集大小＝最大匹配

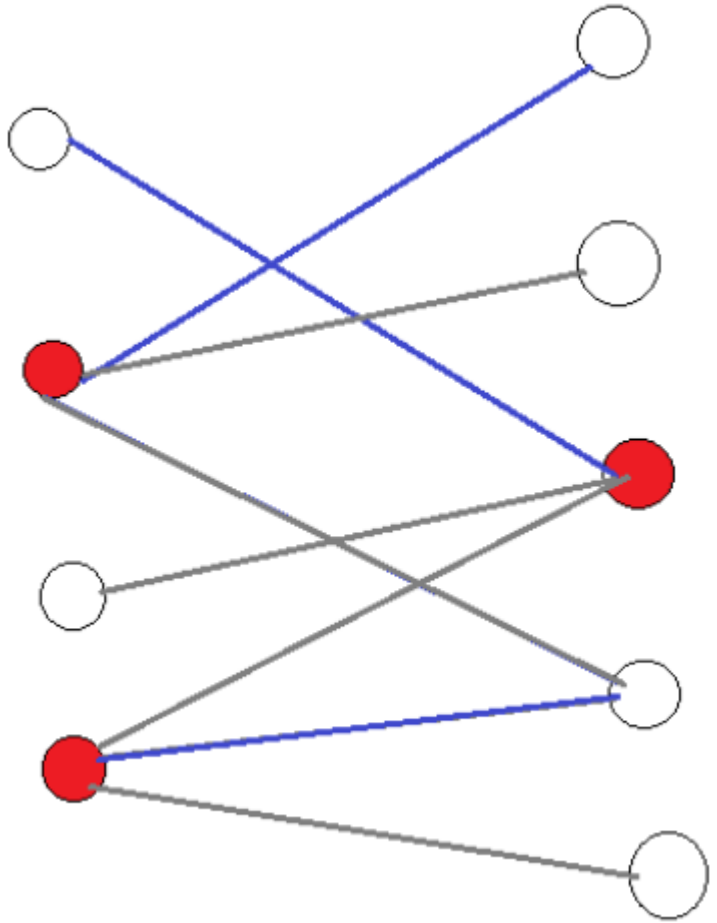
2、选出来的点集可以覆盖所有边

3、最小点覆盖数≥最大匹配数

）

所以选出来的点集即构成最小点覆盖

最小点覆盖⇒最大匹配



3 习题五

7.23

Show that the problem of finding the minimum vertex cover in a bipartite graph reduces to maximum flow. (Hint: Can you relate this problem to the minimum cut in an appropriate network?)

思路三：最小点覆盖 ⇒ 最大匹配 ⇒最大流

最大匹配⇒最大流