    - 854.Floyd求最短路
    - 858.Prim算法求最小生成树
    - 859.Kruskal算法求最小生成树
    - 860.染色法判定二分图
    - 861.二分图的最大匹配
- 第二讲
    - 143.最大异或对
    - 240.食物链
    - 3302.表达式求值
    - 826.单链表
    - 827.双链表
    - 828.模拟栈
    - 829.模拟队列
    - 830.单调栈
    - 836.合并集合
    - 837.连通块中点的数量
    - 838.堆排序
    - 839.模拟堆
    - 840.模拟散列表
    - 841.字符串哈希
- 第五讲
    - 2.01背包问题
    - 282.石子合并
    - 285.没有上司的舞会
    - 291.蒙德里安的梦想
    - 3.完全背包问题
    - 338.计数问题
    - 895.最长上升子序列
    - 897.最长公共子序列
    - 898.数字三角形
    - 899.编辑距离
    - 900.整数划分
    - 901.滑雪
    - 902.最短编辑距离
    - 91.最短Hamilton路径
- 第六讲
    - 104.货仓选址
    - 125.耍杂技的牛

        - 148.合并果子
        - 905.区间选点
        - 906.区间分组
        - 907.区间覆盖
        - 908.最大不相交区间数量
        - 913.排队打水
    - 第四讲
        - 204.表达整数的奇怪方式
        - 866.试除法判定质数
        - 868.筛质数
        - 869.试除法求约数
        - 870.约数个数
        - 871.约数之和
        - 872.最大公约数
        - 873.欧拉函数
        - 874.筛法求欧拉函数
        - 875.快速幂
        - 876.快速幂求逆元
        - 877.扩展欧几里得算法
        - 878.线性同余方程
        - 884.高斯消元解异或线性方程组
        - 885.求组合数I
        - 886.求组合数II
        - 887.求组合数III
        - 888.求组合数IV
        - 889.满足条件的01序列
        - 890.能被整除的数
        - 892.台阶-Nim游戏
        - 893.集合-Nim游戏
        - 894.拆分-Nim游戏
- 算法提高课
    - 第一章动态规划
        - 10.有依赖的背包问题
        - 1010.拦截导弹
        - 1012.友好城市
        - 1013.机器分配
        - 1014.登山
        - 1015.摘花生

Acwing 算法课程目录

目录

- 800.数组元素的目标和
- 801.二进制中1的个数
- 802.区间和
- 803.区间合并

- 第三讲
  - 842.排列数字
  - 843.n-皇后问题
  - 844.走迷宫
  - 845.八数码
  - 846.树的重心
  - 847.图中点的层次
  - 848.有向图的拓扑序列
  - 849.Dijkstra求最短路I
  - 850.Dijkstra求最短路II
  - 851.spfa求最短路
  - 852.spfa判断负环
  - 853.有边数限制的最短路
  - 854.Floyd求最短路
  - 858.Prim算法求最小生成树
  - 859.Kruskal算法求最小生成树
  - 860.染色法判定二分图
  - 861.二分图的最大匹配
- 第二讲

- 1015.摘花生
- 1016.最大上升子序列和
- 1017.怪盗基德的滑翔翼
- 1018.最低通行费
- 1019.庆功会
- 1020.潜水员
- 1021.货币系统
- 1022.宠物小精灵之收服
- 1023.买书
- 1024.装箱问题
- 1027.方格取数
- 1049.大盗阿福
- 1052.设计密码
- 1053.修复DNA
- 1057.股票买卖IV
- 1058.股票买卖V
- 1064.小国王
- 11.背包问题求方案数
- 12.背包问题求具体方案
- 187.导弹防御系统
- 272.最长公共上升子序列
- 275.传纸条
- 278.数字组合

算法基础课

第一讲

2816.判断子序列

```
'''
Author: bddk
Date: 2024-02-19 17:24:41
LastEditors: bddk
LastEditTime: 2024-02-19 17:28:49
'''
n, m = map(int, input().split())
a = list(map(int, input().split()))
b = list(map(int, input().split()))

j = 0
for i in range(m):
    if (a[j] == b[i]):
        j += 1
    if (j == n):
        break


if j == n:
    print("Yes"
          )
else:
    print("No")
```

785.快速排序

```
def quick_sort(l,r,data):
    if l >= r:
        return
    mid = data[(l+r)//2]
    i = l-1
    j = r+1
    while i < j:
        while 1:
            i += 1
            if data[i] >= mid:
                break
        while 1:
            j -= 1
            if data[j] <= mid:
                break
        if i < j:
            data[i],data[j] = data[j],data[i]
    quick_sort(l,j,data)
    quick_sort(j+1,r,data)


n = int(input())
data = [int(x)for x in input().split()]
quick_sort(0,n-1,data)
```

```
print(' '.join(list(map(str,data))))
# print(map(str,data))
```

786.第k个数

```
'''
Author: bddk
Date: 2024-02-12 12:00:35
LastEditors: bddk
LastEditTime: 2024-02-12 13:54:56
'''

n, k = map(int, input().split())
arr = list(map(int, input().split()))


def quick_sort(arr, l, r, k):
    if l == r:
        return arr[l]
    i = l - 1
    j = r + 1
    x = arr[(r + l) // 2]
    while i < j:
        while True:
            i += 1
            if arr[i] >= x:
                break
        while True:
            j -= 1
            if arr[j] <= x:
                break
        if i < j:
            arr[i], arr[j] = arr[j], arr[i]
    if j - l + 1 >= k:
        return quick_sort(arr, l, j, k)
    else:
        return quick_sort(arr, j+1, r, k-(j - l+1))


print(quick_sort(arr, 0, len(arr)-1, k))
```

787.归并排序

```
'''
Author: bddk
Date: 2024-02-13 20:49:59
LastEditors: bddk
LastEditTime: 2024-02-13 21:22:56
'''
```

```python
# '''
# Author: bddk
# Date: 2024-02-13 20:49:59
# LastEditors: bddk
# LastEditTime: 2024-02-13 21:09:41
# '''
n = int(input())
arr = list(map(int, input().split()))


def merge(left, right):
    tmp = []
    l, r = len(left), len(right)
    i = j = 0
    while i < l and j < r:
        if left[i] <= right[j]:
            tmp.append(left[i])
            i += 1
        else:
            tmp.append(right[j])
            j += 1
    while i < l:
        tmp.append(left[i])
        i += 1
    while j < r:
        tmp.append(right[j])
        j += 1
    return tmp


def merge_sort(arr):
    if len(arr) <= 1:
        return arr
    mid = len(arr) // 2
    left = arr[:mid]
    right = arr[mid:]
    left = merge_sort(left)
    right = merge_sort(right)
    return merge(left, right)

result = merge_sort(arr)


print(' '.join(map(str,result)))
```

788.逆序对的数量

```python
'''
Author: bddk
Date: 2024-02-13 21:31:51
LastEditors: bddk
LastEditTime: 2024-02-13 22:12:07
```

```
'''

n = int(input())
arr = list(map(int, input().split()))


def merge_sort(arr):
    if len(arr) <= 1:
        return 0,arr
    mid = len(arr) // 2
    left = arr[:mid]
    right = arr[mid:]
    num_l,left =  merge_sort(left)
    num_r,right =  merge_sort(right)
    num_m,merge_arr = merge(left, right)

    return num_l+num_r+num_m,merge_arr

def merge(left, right):
    tmp = []
    num = 0
    i = j = 0
    l, r = len(left), len(right)
    while i < l and j < r:
        if left[i] <= right[j]:
            tmp.append(left[i])
            i += 1
        else:  # 此处使用else替换第二个if
            tmp.append(right[j])
            num += len(left) - i  # 对于每个right[j], left中剩余的都是逆序对
            j += 1
    # 分别处理剩余元素
    tmp.extend(left[i:])
    tmp.extend(right[j:])

    return num, tmp


num ,tmp = merge_sort(arr)
print(num)
```

789.数的范围

```
'''
Author: bddk
Date: 2024-02-14 18:40:02
LastEditors: bddk
LastEditTime: 2024-02-14 18:49:58
'''
n,q = map(int, input().split())
arr = list(map(int, input().split()))

while q > 0:
    q -= 1
```

```
    x = int(input())
    l ,r = 0, n - 1
    while l < r:
        mid = (l + r) // 2
        if arr[mid] >= x:
            r = mid
        else:
            l = mid + 1
    if arr[l] != x:
         print("-1 -1")
         continue
    left = l
    l ,r = 0, n - 1
    while l < r:
        mid = (l + r + 1) // 2
        if arr[mid] <= x:
            l = mid
        else:
            r = mid - 1
    print(f"{left} {l}")
```

790.数的三次方根

```
'''
Author: bddk
Date: 2024-02-14 19:06:35
LastEditors: bddk
LastEditTime: 2024-02-14 19:16:27
'''
n = float(input())
l, r = -100, 100
while ((r - l) > 1e-8):
    mid = (l+r)/2.0
    if (mid**3 >= n):
        r = mid
    else:
        l = mid + 1e-8
print(f"{l:.6f}")
```

791.高精度加法

```
str1 = int(input())
str2 = int(input())
print(str1 + str2)
```

792.高精度减法

```python
a = int(input())
b = int(input())
print(a-b)
```

793.高精度乘法

```python
a = int(input())
b = int(input())
print(a*b)
```

794.高精度除法

```python
a = int(input())
b = int(input())
print(a//b)
print(a % b)
```

795.前缀和

```python
'''
Author: bddk
Date: 2024-02-14 20:03:40
LastEditors: bddk
LastEditTime: 2024-02-14 20:44:37
'''
n,m = map(int,input().split())
arr = [0] + list(map(int, input().split()))
s = [0] * (len(arr))

for i in range(1,n + 1):
    s[i] = s[i - 1] + arr[i]
while m > 0:
    m -= 1
    l,r = map(int,input().split())
    print(s[r] - s[l -1])
```

796.子矩阵的和

```python
'''
Author: bddk
```

```
Date: 2024-02-14 21:09:08
LastEditors: bddk
LastEditTime: 2024-02-14 21:12:18
'''
n,m,q = map(int,input().split())
arr = [[0]*(m+1) for _ in range(n+1)]
s = [[0]*(m+1) for _ in range(n+1)]

for i in range(1,n+1):
    arr[i] = [0] + list(map(int,input().split()))

for i in range(1,n+1):
    for j in range(1,m+1):
        s[i][j] = s[i-1][j] + s[i][j-1] - s[i-1][j-1] + arr[i][j]

for _ in range(q):
    x1,y1,x2,y2 = map(int,input().split())
    print(s[x2][y2] - s[x1-1][y2] - s[x2][y1-1] + s[x1-1][y1-1])
```

797.差分

```
'''
Author: bddk
Date: 2024-02-14 21:17:08
LastEditors: bddk
LastEditTime: 2024-02-17 14:38:44
'''
n, m = map(int, input().split())
arr = [0] + list(map(int, input().split()))
arr2 = [0] * (n+2)


def insert(l, r, c):
    arr2[l] += c
    arr2[r+1] -= c


for i in range(1, n+1):

    insert(i, i, arr[i])

while m > 0:
    m -= 1
    l, r, c = map(int, input().split())
    insert(l,r,c)


for i in range(1, n+1):
    arr2[i] += arr2[i-1]

print(' '.join(map(str, arr2[1: -1])))
```

798.差分矩阵

```
'''
Author: bddk
Date: 2024-02-17 22:24:32
LastEditors: bddk
LastEditTime: 2024-02-17 22:47:10
'''


n, m, q = map(int, input().split())
a = [[0] * (m + 2) for _ in range(n+2)]
b = [[0] * (m + 2) for _ in range(n+2)]


def insert(x1, y1, x2, y2, c):
    b[x1][y1] += c
    b[x1][y2+1] -= c
    b[x2+1][y1] -= c
    b[x2+1][y2+1] += c


for i in range(1, n+1):
    a[i] = [0] + list(map(int, input().split())) + [0]

for i in range(1, n+1):
    for j in range(1, m+1):
        insert(i, j, i, j, a[i][j])


while q > 0:
    q -= 1
    x1, y1, x2, y2, c = map(int, input().split())
    insert(x1, y1, x2, y2, c)


for i in range(1, n+1):
    for j in range(1, m+1):
        b[i][j] += b[i - 1][j] + b[i][j - 1] - b[i - 1][j - 1]

for i in range(1, n+1):
    print(' '.join(map(str, b[i][1:m+1])))
```

799.最长连续不重复子序列

```
'''
Author: bddk
Date: 2024-02-17 23:28:08
LastEditors: bddk
LastEditTime: 2024-02-17 23:51:47
```

```
'''
n = int(input())
q = list(map(int, input().split()))
s = [0] * (int(1e6)+10)

res = 0

j = 0
for i in range(n):
    s[q[i]] +=1
    while j < i and s[q[i]] > 1:
        s[q[j]] -=1
        j +=1
    res = res if res > i - j +1  else i - j +1

print(res)
```

800.数组元素的目标和

```
'''
Author: bddk
Date: 2024-02-18 18:54:02
LastEditors: bddk
LastEditTime: 2024-02-18 19:11:13
'''
n,m,x = map(int,input().split())
a = list(map(int,input().split()))
b = list(map(int,input().split()))


i , j = 0, m -1
for i in range(n):
    while j < m and a[i] + b[j] > x:
        j -=1
    if a[i] + b[j] == x:
        break
    if a[i] + b[j] < x:
        continue

print(f"{i} {j}")
```

801.二进制中1的个数

```
'''
Author: bddk
Date: 2024-02-19 17:36:40
LastEditors: bddk
LastEditTime: 2024-02-19 17:36:44
'''
```

```python
n = int(input())
a = list(map(int, input().split()))

for i in range(n):
    res = 0
    while a[i]:
        if a[i] % 2 ==1:
            res += 1
        a[i] >>= 1

    print(res, end=" ")
```

802.区间和

```python
'''
Author: bddk
Date: 2024-02-27 11:11:07
LastEditors: bddk
LastEditTime: 2024-02-27 11:36:25
'''
add = []
query = []
IndexList = []

n, m = map(int, input().split())


for i in range(n):
    add.append(list(map(int, input().split())))
    IndexList.append(add[i][0])

for i in range(m):
    query.append(list(map(int, input().split())))
    IndexList.append(query[i][0])
    IndexList.append(query[i][1])

IndexList.sort()
IndexList = list(set(IndexList))

num = [0] * (len(IndexList)+1)
s = [0]*(len(IndexList)+1)


def find(x):
    l = 0
    r = len(IndexList)-1
    while l < r:
        mid = (l+r)//2
        if IndexList[mid] >= x:
            r = mid
        else:
            l = mid + 1
    return r + 1
```

```
for i in range(len(add)):
    x = find(add[i][0])
    num[x] += add[i][1]

for i in range(len(num)):
    s[i] = s[i - 1] + num[i]

for i in range(len(query)):
    l = find(query[i][0])
    r = find(query[i][1])
    print(s[r] - s[l - 1])
```

803.区间合并

```
'''
Author: bddk
Date: 2024-02-27 11:55:35
LastEditors: bddk
LastEditTime: 2024-02-27 19:29:20
'''
segs = []
n = int(input())
for i in range(n):
    segs.append(list(map(int, input().split())))


segs.sort(key=lambda x: x[0])


def merge(segs):
    res = []
    l, r = -1e9, -1e9
    for seg in segs:
        if r < seg[0]:
            if l != -1e9:
                res.append([l,r])
            l, r = seg[0], seg[1]
        else:
            r = r if r > seg[1] else seg[1]
    if l != -1e9:
        res.append([l, r])
    return res


segs = merge(segs)



print(len(segs))
```

第三讲

842.排列数字

```
'''
Author: bddk
Date: 2024-03-05 20:41:06
LastEditors: bddk
LastEditTime: 2024-03-05 20:41:11
'''
n = int(input())
p = [0]*(n)
flag = [0]*(n+1)


def dfs(u):
    if u == n:
        print(' '.join(map(str, p)))
        return
    for i in range(1, n+1):
        if not flag[i]:
            flag[i] = 1
            p[u] = i
            dfs(u + 1)
            flag[i] = 0


dfs(0)
```

843.n-皇后问题

```
'''
Author: bddk
Date: 2024-03-13 14:55:15
LastEditors: bddk
LastEditTime: 2024-03-13 15:28:47
'''
n = int(input())

N = 10
set = [['.'] * N for _ in range(N)]
row = [0] * N *2
dg = [0] * N *2
udg = [0] * N *2

def queen(i):
    if i == n:
```

```
        for i in range(n):
            for j in range(n):
                print(set[i][j],end = '')
            print()
        print()
        return
    for j in range(n):
        if row[j] == 0 and dg[i+j] == 0 and udg[i-j+n] == 0:
            row[j] = 1
            dg[i+j] = 1
            udg[i-j+n] = 1
            set[i][j] = 'Q'
            queen(i+1)
            row[j] = 0
            dg[i+j] = 0
            udg[i-j+n] = 0
            set[i][j] = '.'

queen(0)
```

844.走迷宫

```
'''
Author: bddk
Date: 2024-03-15 10:52:36
LastEditors: bddk
LastEditTime: 2024-03-15 11:15:31
'''
from collections import deque

N = 105
d = [[-1]*N for _ in range(N)]
dx = [1, 0, -1, 0]
dy = [0, 1, 0, -1]

n, m = map(int, input().split())
g = [list(map(int,input().split())) for _ in range(n)]

def bfs():
    q = deque()
    q.append((1,1))
    d[1][1] = 0
    while q:
        x, y = q.popleft()
        if x == n and y == m:
            break
        for i in range(4):
            a = x + dx[i]
            b = y + dy[i]
            if a >= 1 and a <= n and b >= 1 and b <= m and d[a][b] == -1 and
g[a - 1][b - 1] == 0:
                d[a][b] = d[x][y] + 1
                q.append((a, b))
```

```
bfs()
print(d[n][m])
```

845.八数码

```
'''
Author: bddk
Date: 2024-03-15 15:05:07
LastEditors: bddk
LastEditTime: 2024-04-09 16:10:03
'''
from collections import deque


def bfs(state):
    end = '12345678x'
    q = deque()
    d = {state: 0}
    q.append(state)
    dx = [1, 0, -1, 0]
    dy = [0, 1, 0, -1]
    while q:
        now = q.popleft()
        distance = d[now]
        if now == end:
            print(distance)
            return
        t = now.find('x')
        x = t // 3
        y = t % 3
        for i in range(4):
            a = x + dx[i]
            b = y + dy[i]
            if a >= 0 and a < 3 and b >= 0 and b < 3:
                p = a * 3 + b
                next = list(now)
                next[t], next[p] = next[p], next[t]
                next = ''.join(next)
                if next not in d:
                    d[next] = distance + 1
                    q.append(next)
                next = list(now)
                next[t], next[p] = next[p], next[t]
                next = ''.join(next)
    print(-1)


n = input().split()
str = ''
for i in n:
    str += i
```

```
bfs(str)
```

846.树的重心

```
N = int(1e5 + 10)
M = 2 * N

h = [-1] * N
e = [-1] * M
ne = [-1] * M
idx = 0
st = [0] * N
ans = N


def add(a, b):
    global idx
    e[idx] = b
    ne[idx] = h[a]
    h[a] = idx
    idx += 1


def dfs(u):
    global ans
    st[u] = 1

    size = 0
    SUM = 0

    i = h[u]
    while i != -1:
        j = e[i]
        if st[j]:
            i = ne[i]
            continue
        s = dfs(j)
        size = max(size, s)
        SUM += s

        i = ne[i]

    size = max(size, n - SUM - 1)
    ans = min(ans, size)

    return SUM + 1


n = int(input())
for _ in range(n - 1):
    a, b = map(int, input().split())
    add(a, b)
    add(b, a)
```

```
dfs(1)

print(ans)
```

847.图中点的层次

```
from queue import Queue

N = int(1e5) + 10
M = 2 * N
h = [-1] * N
e = [0] * M
ne = [-1] * M
idx = 0
q = Queue()
d = [-1] * N


def add(a, b):
    global idx
    e[idx] = b
    ne[idx] = h[a]
    h[a] = idx
    idx += 1


def bfs():
    q.put(1)
    d[1] = 0
    while not q.empty():
        t = q.get()

        i = h[t]
        while i != - 1:
            j = e[i]

            if d[j] == -1:
                d[j] = d[t] + 1
                q.put(j)
            i = ne[i]


n, m = map(int, input().split())

for _ in range(m):
    a, b = map(int, input().split())
    add(a, b)

bfs()
```

```
print(d[n])
```

848.有向图的拓扑序列

```
from queue import Queue

N = int(1e5) + 10

e = [0] * N
ne = [0] * N
h = [-1] * N
d = [0] * N
idx = 0
top = [0] * N


def add(a, b):
    global idx
    e[idx] = b
    ne[idx] = h[a]
    h[a] = idx
    idx += 1
    d[b] += 1


def topsort():
    cnt = 0
    q = Queue()
    for i in range(1, n + 1):
        if d[i] == 0:
            q.put(i)
    while not q.empty():
        t = q.get()
        top[cnt] = t
        cnt += 1

        i = h[t]
        while i != -1:
            j = e[i]
            d[j] -= 1
            if d[j] == 0:
                q.put(j)
            i = ne[i]
    return cnt == n


n, m = map(int, input().split())

for _ in range(m):
    a, b = map(int, input().split())
    add(a, b)

if topsort():
```

```
    print(" ".join(map(str, top[:n])))
else:
    print(-1)
```

849.Dijkstra求最短路I

```
N = 510

INF = float("inf")

g = [[INF for _ in range(N)] for _ in range(N)]

for i in range(N):
    for j in range(N):
        if i == j:
            g[i][j] = 0


d = [INF for _ in range(N)]

d[1] = 0

st = [0] * N


def dijkstra():
    for _ in range(n):
        t = -1
        for i in range(1, n + 1):
            if st[i] == 0 and (t == -1 or d[t] > d[i]):
                t = i
        st[t] = 1

        for i in range(1, n + 1):
            d[i] = min(d[i], d[t] + g[t][i])
    if d[n] == INF:
        return -1
    else:
        return d[n]


n, m = map(int, input().split())

for _ in range(m):
    a, b, c = map(int, input().split())
    g[a][b] = min(g[a][b], c)  # 存在重边，所以的当2个点之前第二次赋值时需要判断和前一次的赋值的大小
```

```
print(dijkstra())
```

850.Dijkstra求最短路II

```
from queue import PriorityQueue

INF = float("inf")
N = int(1e6) + 10
e = [0]*N
ne = [0]*N
h = [-1]*N
st = [0]*N
w = [0]*N
idx = 0
d = [INF for _ in range(N)]


def add(a, b, c):
    global idx

    e[idx] = b
    ne[idx] = h[a]
    h[a] = idx
    w[idx] = c
    idx += 1



def dijkstra():

    d[1] = 0
    pq = PriorityQueue()
    pq.put((0, 1))

    while not pq.empty():
        dis, ver = pq.get()

        if st[ver]:
            continue
        st[ver] = 1

        t = h[ver]

        i = t
        while i != -1:
            j = e[i]

            if d[j] > d[ver] + w[i]:
                d[j] = d[ver] + w[i]
                pq.put((d[j], j))
            i = ne[i]
    if d[n] == INF:
        return -1
```

```
    else:
        return d[n]


n, m = map(int, input().split())

for _ in range(m):
    a, b, c = map(int, input().split())
    add(a, b, c)

print(dijkstra())
```

851.spfa求最短路

```
from queue import Queue

N = int(1e5)+10

INF = float("inf")

e = [0] * N
ne = [0] * N
h = [-1] * N
d = [INF] * N
idx = 0
w = [0] * N  # 标记这个点在不在队列中
st = [0] * N


def add(a, b, c):
    global idx
    e[idx] = b
    ne[idx] = h[a]
    h[a] = idx
    w[idx] = c
    idx += 1


def spfa():
    d[1] = 0
    q = Queue()
    q.put(1)
    st[1] = 1

    while not q.empty():
        t = q.get()
        st[t] = 0
        i = h[t]

        while i != -1:

            j = e[i]
```

```
            if d[j] > d[t] + w[i]:
                d[j] = d[t] + w[i]
                if st[j] == 0:
                    q.put(j)
                    st[j] = 1

            i = ne[i]

    if d[n] == INF:
        return "impossible"
    else:
        return d[n]


n, m = map(int, input().split())

for _ in range(m):
    a, b, c = map(int, input().split())
    add(a, b, c)

print(spfa())
```

852.spfa判断负环

```
from collections import deque

N = 2000 + 10
M = 10000 + 10
INF = 0x3f3f3f3f # 注意这里不能用float("inf"),因为python中的inf无论减去多少个数，都是inf,不能到达负无穷，这道题需要d数组能减到负值

e = [0] * M
ne = [0] * M
h = [-1] * N
idx = 0
d = [INF] * N
cnt = [0] * N
st = [0] * N
w = [0] * M


def add(a, b, c):
    global idx
    e[idx] = b
    w[idx] = c
    ne[idx] = h[a]
    h[a] = idx
    idx += 1


def spfa():
    d[1] = 0
    q = deque()
```

```
    for i in range(1, n + 1):
        q.append(i)
        st[i] = 1 # 所有点入队列是因为从起点不一定能到负权边

    while q:
        t = q.popleft()
        st[t] = 0
        i = h[t]

        while i != -1:
            j = e[i]

            if d[j] > d[t] + w[i]:
                d[j] = d[t] + w[i]
                cnt[j] = cnt[t] + 1
                if cnt[j] >= n:
                    return "Yes"
                if st[j] == 0:
                    st[j] = 1
                    q.append(j)
            i = ne[i]

    return "No"


n, m = map(int, input().split())

for _ in range(m):
    a, b, c = map(int, input().split())
    add(a, b, c)


print(spfa())
```

853.有边数限制的最短路

```
INF = float("inf")
N = 510


dist = [INF for _ in range(N)]
backup = [INF for _ in range(N)]


edges = []


def bellman_ford():
    dist[1] = 0
    for _ in range(k):
        backup = dist.copy()
        for i in range(m):
            a, b, c = edges[i]
```

```
            if dist[b] > backup[a] + c:
                dist[b] = backup[a] + c
    if dist[n] == INF:
        return "impossible"
    else:
        return dist[n]


n, m, k = map(int, input().split())

for _ in range(m):
    a, b, c = map(int, input().split())
    edges.append((a, b, c))

print(bellman_ford())
```

854.Floyd求最短路

```
N = 210
M = 20000 + 10

INF = float("inf")

d = [[INF for _ in range(N)] for _ in range(N)]


def floyd():
    for k in range(1, n + 1):
        for i in range(1, n+1):
            for j in range(1, n+1):
                d[i][j] = min(d[i][j], d[i][k] + d[k][j])


n, m, q = map(int, input().split())

for i in range(1,n + 1):
    d[i][i] = 0


for _ in range(m):
    a, b, c = map(int, input().split())
    d[a][b] = min(d[a][b], c)

floyd()


for _ in range(q):
    a, b = map(int, input().split())
    if d[a][b] == INF:
        print("impossible")
    else:
```

```
        print(d[a][b])
```

858.Prim算法求最小生成树

```
N = 510
M = int(1e5) + 10

INF = float("inf")

g = [[INF for _ in range(N)] for _ in range(N)]
d = [INF] * N
st = [0] * N


# def prim():
#     res = 0
#     d[1] = 0
#     for j in range(n):
#         t = -1
#         for i in range(1, n+1):
#             if st[i] == 0 and (t == -1 or d[t] > d[i]):
#                 t = i

#         st[t] = 1

#         if j and d[t] == INF:
#             return "impossible"

#         res += d[t]

#         for i in range(1, n+1):
#             d[i] = min(d[i], g[t][i])

#     return res
def prim():
    res = 0
    d[1] = 0
    for j in range(n):
        t = -1
        for i in range(1, n+1):
            if st[i] == 0 and (t == -1 or d[t] > d[i]):
                t = i

        st[t] = 1

        if j and d[t] == INF:
            return "impossible"

        for i in range(1, n+1):
            if st[i] == 0:
                d[i] = min(d[i], g[t][i])
        res += d[t]
    return res
```

```
n, m = map(int, input().split())

for i in range(1, n + 1):
    g[i][i] = 0


for _ in range(m):
    a, b, c = map(int, input().split())
    g[a][b] = g[b][a] = min(g[a][b], c)

print(prim())
```

859.Kruskal算法求最小生成树

```
N = int(1e5) + 10
M= int(2e5) + 10

p = [-1] * N

edges = []

def find(x):
    if p[x] != x:
        p[x] = find(p[x])
    return p[x]

n,m = map(int,input().split())

for _ in range(m):
    a,b,c = map(int,input().split())
    edges.append((a,b,c))

for i in range(1,n + 1):
    p[i] = i

edges.sort(key= lambda x:x[2])

res = 0
cnt = 0
for i in range(m):
    a,b,c = edges[i]
    a = find(a)
    b = find(b)
    if a!= b:
       p[a] = b
       res += c
       cnt += 1

if cnt == n - 1:
    print(res)
else:
```

```
    print("impossible")
```

860.染色法判定二分图

```
from collections import deque

N = int(1e5) + 10
M = int(2e5) + 10

h = [-1] * N
e = [0] * M
ne = [0] * M
idx = 0
color = [0] * N
flag = 0


def add(a, b):
    global idx
    e[idx] = b
    ne[idx] = h[a]
    h[a] = idx
    idx += 1

# python的dfs会爆栈
# def dfs(u, c):
#     color[u] = c

#     i = h[u]

#     while i != -1:
#         j = e[i]

#         if color[j] == 0:
#             # color[j] = 3 -c
#             if  dfs(j, 3-c) == False:
#                 return False
#         else:
#             if color[j] == c:
#                 return False

#         i = ne[i]

#     return True

def bfs(u,c):
    q = deque()

    q.append(u)
    color[u] = c

    while q:
        t = q.popleft()
```

```
        i = h[t]

        while i != -1:
            j = e[i]

            if color[j] == 0:
                color[j] = 3 -color[t]
                q.append(j)
            else:
                if color[j] == color[t]:
                    return False

            i = ne[i]

    return True

n, m = map(int, input().split())

for _ in range(m):
    a, b = map(int, input().split())
    add(a, b)
    add(b, a)

flag = True

for i in range(1, n+1): # 有可能是非联通图,for循环遍历每个非联通图
    if color[i] == 0:
        flag = bfs(i,1)
        if flag == False:
            break


if flag:
    print("Yes")
else:
    print("No")
```

861.二分图的最大匹配

```
N = 510
M = int(1e5) + 10

e = [0] * M
ne = [0] * M
h = [-1] * N
idx = 0
st = [0] * N
match = [0] * N


def add(a, b):
    global idx
```

```
    e[idx] = b
    ne[idx] = h[a]
    h[a] = idx
    idx += 1


def find(u):
    i = h[u]
    while i != -1:
        j = e[i]

        if st[j] == 0: # 如果没有将 st[j] 设置为 1，那么在递归中，顶点 j 就有可能被重新访问，造成无限递归。
            st[j] = 1
            if match[j] == 0 or find(match[j]):
                match[j] = u
                return True

        i = ne[i]
    return False


n1, n2, m = map(int, input().split())

for _ in range(m):
    a, b = map(int, input().split())
    add(a, b)

res = 0
for i in range(1, n1 + 1):

    for j in range(N):
        st[j] = 0

    if find(i):
        res += 1

print(res)
```

第二讲

143.最大异或对

```
'''
Author: bddk
Date: 2024-03-03 20:04:16
LastEditors: bddk
LastEditTime: 2024-03-03 20:38:01
'''
N = 3000000
n = int(input())
```

```python
arr = list(map(int, input().split()))

son = [[0] * 2 for _ in range(N)]
idx = 0


def insert(x):
    global idx
    q = 0
    for i in range(30, -1, -1):
        u = x >> i & 1
        if son[q][u] == 0:
            idx += 1
            son[q][u] = idx
        q = son[q][u]


def query(x):
    global idx
    q = 0
    res = 0
    for i in range(30, -1, -1):
        u = x >> i & 1
        if son[q][~u]:
            res += 1 << i
            q = son[q][~u]
        else:
            q = son[q][u]
    return res



res = 0
for i in range(n):
    insert(arr[i])
    res = res if res > query(arr[i]) else query(arr[i])

print(res)
```

240.食物链

```python
'''
Author: bddk
Date: 2024-03-04 20:22:18
LastEditors: bddk
LastEditTime: 2024-03-04 20:22:22
'''
n, k = map(int, input().split())
p = [i for i in range(n+1)]
d = [0]*(n+1)


def find(x):
```

```
    # while p[x] != x:
    #     d[x] += d[p[x]]
    #     p[x] = p[p[x]]
    #     x = p[x]
    # return x
    if p[x] != x:
        t = find(p[x])
        d[x] += d[p[x]]
        p[x] = t
    return p[x]


res = 0

for _ in range(k):
    D, x, y = map(int, input().split())
    if x > n or y > n:
        res += 1
        continue
    if D == 1:
        px, py = find(x), find(y)
        if px == py and (d[x] - d[y])%3 != 0:
            res += 1
        elif px != py:
            p[px] = py
            d[px] = d[y] - d[x]
    else:
        px, py = find(x), find(y)
        if px == py and (d[x] - d[y] - 1)%3 != 0:
            res += 1
        elif px != py:
            p[px] = py
            d[px] = d[y] - d[x] + 1

print(res)
```

3302.表达式求值

```
'''
Author: bddk
Date: 2024-02-28 23:09:16
LastEditors: bddk
LastEditTime: 2024-02-29 15:29:05
'''
nums = []
ops = []

pr = {'(': 0, '+': 1, '-': 1, '*': 2, '/': 2}


def new_eval():
    a = nums.pop()
    b = nums.pop()
    op = ops.pop()
```

```
    if op == '+':
        nums.append(a + b)
    elif op == '-':
        nums.append(b - a)
    elif op == '*':
        nums.append(a * b)
    elif op == '/':
        nums.append(int(b / a))


str = input()

i = 0
while i < len(str):
    if str[i].isdigit():
        j = i
        num = 0
        while j < len(str) and str[j].isdigit():
            num = num*10 + int(str[j])
            j += 1
        nums.append(num)
        i = j - 1
    elif str[i] == '(':
        ops.append(str[i])
    elif str[i] == ')':
        while len(ops) > 0 and ops[-1] != '(':
            new_eval()
        ops.pop()
    else:
        if len(ops) > 0 and pr[str[i]] <= pr[ops[-1]]:
            new_eval()
        ops.append(str[i])
    i += 1

while len(ops) > 0:
    new_eval()
print(nums[-1])
```

826.单链表

```
'''
Author: bddk
Date: 2024-02-28 14:21:08
LastEditors: bddk
LastEditTime: 2024-02-28 14:56:41
'''
N = 100010
head = -1
idx = 0

m = int(input())
op = []
```

```python
e = [0]*N
ne = [0]*N

for _ in range(m):
    op.append(list(map(str, input().split())))


def add_to_head(x):
    global idx, head
    e[idx] = x
    ne[idx] = head
    head = idx
    idx += 1


def delete(k):
    global head
    if k == 0:  # 删除头节点
        head = ne[head]
    else:
        ne[k-1] = ne[ne[k-1]]  # 删除非头节点


def insert(k, x):
    global idx
    e[idx] = x
    ne[idx] = ne[k - 1]
    ne[k - 1] = idx
    idx += 1


for i in range(m):
    if op[i][0] == 'H':
        add_to_head(int(op[i][1]))
    if op[i][0] == 'D':
        delete(int(op[i][1]))
    if op[i][0] == 'I':
        insert(int(op[i][1]), int(op[i][2]))

while head != -1:
    print(e[head], end=' ')
    head = ne[head]
```

827.双链表

```python
'''
Author: bddk
Date: 2024-02-28 19:01:14
LastEditors: bddk
LastEditTime: 2024-02-28 21:00:53
'''
N = 100010
m = int(input())
```

```
e = [0]*N
l = [0]*N
r = [0]*N

r[0] = 1
l[1] = 0
idx = 2


def insert(k, x):
    global idx
    e[idx] = x
    r[idx] = r[k]
    l[idx] = k
    l[r[k]] = idx
    r[k] = idx
    idx += 1


def remove(k):
    r[l[k]] = r[k]
    l[r[k]] = l[k]


while m:
    m -= 1
    op = list(map(str, input().split()))
    if op[0] == 'L':
        insert(0, int(op[1]))
    if op[0] == 'R':
        insert(l[1], int(op[1]))
    if op[0] == 'D':
        remove(int(op[1]) + 1)
    if op[0] == 'IL':
        insert(l[int(op[1])+1], int(op[2]))
    if op[0] == 'IR':
        insert(int(op[1])+1, int(op[2]))

while r[0] != 1:
    print(e[r[0]], end=' ')
    r[0] = r[r[0]]
```

828.模拟栈

```
'''
Author: bddk
Date: 2024-02-28 22:44:06
LastEditors: bddk
LastEditTime: 2024-02-28 22:53:06
'''
stack = []
```

```
def push(x):
    stack.append(x)


def pop():
    return stack.pop()


def empty():
    if not stack:
        return 'YES'
    return 'NO'


def query():
    return stack[-1]


m = int(input())

while m:
    m -= 1
    op = input().split()
    if op[0] == 'push':
        push(op[1])
    if op[0] == 'pop':
        op()
    if op[0] == 'empty':
        print(empty())
    if op[0] == 'query':
        print(query())
```

829.模拟队列

```
'''
Author: bddk
Date: 2024-02-29 15:47:48
LastEditors: bddk
LastEditTime: 2024-02-29 15:47:53
'''
queue = []

def push(x):
    queue.append(x)
def pop():
    queue.pop(0)
def empty():
    return len(queue) == 0
def query():
    return queue[0]

m = int(input())
```

```
for _ in range(m):
    op = input().split()
    if op[0] == 'push':
        push(int(op[1]))
    if op[0] == 'pop':
        pop()
    if op[0] == 'query':
        print(query())
    if op[0] == 'empty':
        if empty():
            print('YES')
        else:
            print('NO')
```

830.单调栈

```
'''
Author: bddk
Date: 2024-02-29 16:13:02
LastEditors: bddk
LastEditTime: 2024-02-29 16:26:34
'''
stack = []

n = int(input())
list = list(map(int, input().split()))


i = 0
while i < n:
    while (stack and stack[-1] >= list[i]):
        stack.pop()
    if not stack:
        print("-1", end=" ")
    else:
        print(stack[-1], end=" ")
    stack.append(list[i])

    i += 1
```

836.合并集合

```
'''
Author: bddk
Date: 2024-03-04 18:33:06
LastEditors: bddk
LastEditTime: 2024-03-04 19:02:45
'''
```

```
n, m = map(int, input().split())

p = [i for i in range(n + 1)]



def find(x):
    while p[x] != x:
        p[x] = p[p[x]]
        x = p[x]
    return p[x]


for _ in range(m):
    op = input().split()
    a, b = int(op[1]), int(op[2])
    if op[0] == 'M':
        p[find(a)] = find(b)
    else:
        if find(a) == find(b):
            print('Yes')
        else:
            print('No')
```

837.连通块中点的数量

```
'''
Author: bddk
Date: 2024-03-04 19:03:16
LastEditors: bddk
LastEditTime: 2024-03-04 19:28:01
'''
'''
Author: bddk
Date: 2024-03-04 18:33:06
LastEditors: bddk
LastEditTime: 2024-03-04 19:02:45
'''
n, m = map(int, input().split())

p = [i for i in range(n + 1)]
cnt = [1] * (n + 1)


def find(x):
    while p[x] != x:
        p[x] = p[p[x]]
        x = p[x]
    return p[x]



for _ in range(m):
```

```
    op = input().split()

    if op[0] == 'C':
        a, b = int(op[1]), int(op[2])
        p_a = find(a)
        p_b = find(b)
        if p_a != p_b:
            p[p_a] = p_b
            cnt[p_b] += cnt[p_a]
    elif op[0] == 'Q1':
        a, b = int(op[1]), int(op[2])
        if find(a) == find(b):
            print('Yes')
        else:
            print('No')
    else:
        if op[0] == 'Q2':
            a = int(op[1])
            print(cnt[find(a)])
```

838.堆排序

```
'''
Author: bddk
Date: 2024-03-04 22:55:27
LastEditors: bddk
LastEditTime: 2024-03-04 23:20:24
'''
# import heapq

# n,m = map(int,input().split() )
# arr = list(map(int,input().split()))

# heapq.heapify(arr)

# for _ in range(m):
#    print(heapq.heappop(arr),end=' ')

n, m = map(int, input().split())
arr = [0] + list(map(int, input().split()))

cnt = n

def down(x):
    u = x
    if x*2 <= cnt and arr[u] > arr[x*2]:
        u = x*2
    if x*2+1 <= cnt and arr[u] > arr[x*2+1]:
        u = x*2+1
    if u != x:
        arr[u], arr[x] = arr[x], arr[u]
        down(u)
```

```
for i in range(n//2, 0, -1):
    down(i)

for _ in range(m):
    print(arr[1], end=' ')
    arr[1] = arr[cnt]
    cnt -= 1
    down(1)
```

839.模拟堆

```
'''
Author: bddk
Date: 2024-03-04 23:52:36
LastEditors: bddk
LastEditTime: 2024-03-05 00:13:47
'''
N = 100010
h = [0] * (N + 1)
ph = [0] * (N + 1)
hp = [0] * (N + 1)
cnt = m = 0


def swap(a, b):
    ph[hp[a]], ph[hp[b]] = ph[hp[b]], ph[hp[a]]
    hp[a], hp[b] = hp[b], hp[a]
    h[a], h[b] = h[b], h[a]


def up(x):
    while x//2 > 0 and h[x] < h[x//2]:
        swap(x, x//2)
        x = x//2


def down(x):
    u = x
    if x*2 <= cnt and h[u] > h[x*2]:
        u = x*2
    if x*2+1 <= cnt and h[u] > h[x*2+1]:
        u = x*2+1
    if u != x:
        swap(x, u)
        down(u)


n = int(input())

for _ in range(n):
    op = input().split()
    if op[0] == 'I':
```

```
        x = int(op[1])
        cnt += 1
        m += 1
        h[cnt] = x
        ph[m] = cnt
        hp[cnt] = m
        up(cnt)
    if op[0] == 'PM':
        print(h[1])
    if op[0] == 'DM':
        swap(1, cnt)
        cnt -= 1
        down(1)
    if op[0] == 'D':
        k = int(op[1])
        k = ph[k]
        swap(k, cnt)
        cnt -= 1
        up(k)
        down(k)
    if op[0] == 'C':
        k, x = int(op[1]), int(op[2])
        k = ph[k]
        h[k] = x
        up(k)
        down(k)
```

840.模拟散列表

```
'''
Author: bddk
Date: 2024-03-05 13:12:21
LastEditors: bddk
LastEditTime: 2024-03-05 13:14:27
'''
# from collections import defaultdict

# dict = defaultdict(int)
# n = int(input())

# for _ in range(n):
#     op = input().split()
#     x = int(op[1])
#     if op[0] == 'I':
#         dict[x] += 1
#     if op[0] == 'Q':
#         if dict[x]:
#             print('Yes'
#                   )
#         else:
#             print('No')
N = 200003
null = 0x3f3f3f3f
```

```
h = [null] * N


def find(x):
    t = x % N
    while h[t] != null and h[t] != x:
        t += 1
        if t == N:
            t = 0
    return t


n = int(input())

for _ in range(n):
    op = input().split()
    x = int(op[1])
    if op[0] == 'I':
        h[find(x)] = x
    if op[0] == 'Q':
        if h[find(x)] != null:
            print('Yes'
                  )
        else:
            print('No')
```

841.字符串哈希

```
'''
Author: bddk
Date: 2024-03-05 13:25:11
LastEditors: bddk
LastEditTime: 2024-03-05 13:27:27
'''
N = 100010
P = 131
Q = 1 << 64
h = [0]*N
p = [0]*N

def find(l,r):
    return ((h[r]-h[l-1]*p[r-l+1])%Q)

n, m = map(int, input().split())
str = ' ' + input()

p[0] = 1
for i in range(1,n+1):
    p[i] = (p[i-1]*P)%Q
    h[i] = (h[i-1]*P+ord(str[i]))%Q

for _ in range(m):
    l1,r1,l2,r2 = map(int, input().split())
    if find(l1,r1) == find(l2,r2):
```

```
        print("Yes")
    else:
        print("No")
```

第五讲

2.01背包问题

```
# N = 1010
# V = N * [0]
# W = N * [0]
# f = [[0 for _ in range(N)] for _ in range(N)]

# n, v = map(int, input().split())

# for i in range(1, n + 1):
#     vi, wi = map(int, input().split())
#     V[i] = vi
#     W[i] = wi

# for i in range(1, n + 1):
#     for j in range(1, v + 1):
#         f[i][j] = f[i - 1][j]
#         if j >= V[i]:
#             f[i][j] = max(f[i][j], f[i - 1][j - V[i]] + W[i])

# print(f[n][v])
N = 1010
V = N * [0]
W = N * [0]
f = [0 for _ in range(N)]
n, v = map(int, input().split())

for i in range(1, n + 1):
    vi, wi = map(int, input().split())
    V[i] = vi
    W[i] = wi

for i in range(1, n + 1):
    for j in range(v, 0, -1):
        if j >= V[i]:
            f[j] = max(f[j], f[j - V[i]] + W[i])

print(f[v])
```

282.石子合并

```
N = 310

f = [[0 for _ in range(N)] for _ in range(N)]

s = [0] * N

n = int(input())

a = [0] + list(map(int, input().split()))


for i in range(n + 1):
    s[i] = s[i - 1] + a[i]

for LEN in range(2, n + 1):
    for i in range(1, n + 2 - LEN):
        j = i + LEN - 1
        f[i][j] = int(1e9)
        for k in range(i, j):
            f[i][j] = min(f[i][j], f[i][k] + f[k + 1][j] + s[j] - s[i - 1])

print(f[1][n])
```

285.没有上司的舞会

```
N = 6000 + 10

f = [[0 for _ in range(2)] for _ in range(N)]

e = [0] * N
ne = [0] * N
h = [-1] * N
idx = 0

st = [0] * N
w = [0] * N


def add(a, b):
    global idx

    e[idx] = b
    ne[idx] = h[a]
    h[a] = idx
    idx += 1


def dfs(x):
    f[x][1] = w[x]

    i = h[x]
```

```
    while i != -1:

        t = e[i]

        dfs(t)

        f[x][1] += f[t][0]
        f[x][0] += max(f[t][0], f[t][1])

        i = ne[i]


n = int(input())

for i in range(1, n + 1):
    w[i] = int(input())

for _ in range(n - 1):
    a, b = map(int, input().split())
    add(b, a)
    st[a] = 1

root = 1
while st[root]:
    root += 1

dfs(root)
print(max(f[root][0], f[root][1]))
```

291.蒙德里安的梦想

```
# N = 12
# M = 1 << N

# f = [[0 for _ in range(M)] for _ in range(N)]
# st = [1] * M


# while True:
#     n, m = map(int, input().split())
#     if n == 0 and m == 0:
#         break

#     for i in range(1 << n):
#         is_valid = 1
#         cnt = 0
#         for j in range(n):
#             if (i >> j) & 1:
#                 if cnt & 1:
#                     is_valid = 0
#                     break
#             else:
```

```
#                 cnt += 1
#         if cnt & 1:
#             is_valid = 0
#         st[i] = is_valid
#     f = [[0 for _ in range(M)] for _ in range(N)]
#     f[0][0] = 1
#     for i in range(1, m + 1):
#         for j in range(1 << n):
#             for k in range(1 << n):
#                 if (j & k) == 0 and st[j | k]:
#                     f[i][j] += f[i - 1][k]

#     print(f[m][0])
N = 12
M = 1 << N

f = [[0 for _ in range(M)] for _ in range(N)]
st = [1] * M


while True:
    n, m = map(int, input().split())
    if n == 0 and m == 0:
        break

    for i in range(1 << n):
        is_valid = 1
        cnt = 0
        for j in range(n):
            if (i >> j) & 1:
                if cnt & 1:
                    is_valid = 0
                    break
                cnt = 0
            else:
                cnt += 1
        if cnt & 1:
            is_valid = 0
        st[i] = is_valid
    states = []
    for i in range(1 << n):
        state = []
        for j in range(1 << n):
            if (i & j) == 0 and st[i | j]:
                state.append(j)
        states.append(state)
    f = [[0 for _ in range(M)] for _ in range(N)]
    f[0][0] = 1
    for i in range(1, m + 1):
        for j in range(1 << n):
            for k in states[j]:
                f[i][j] += f[i - 1][k]

    print(f[m][0])
```

3.完全背包问题

```
# N = 1010

# w = [0] * N
# v = [0] * N
# f = [[0 for _ in range(N)]for _ in range(N)]


# n, m = map(int, input().split())

# for i in range(1, n + 1):
#     vi, wi = map(int, input().split())
#     v[i] = vi
#     w[i] = wi

# for i in range(1, n + 1):
#     for j in range(1, m + 1):
#         f[i][j] = f[i - 1][j]
#         if j >= v[i]:
#             f[i][j] = max(f[i][j], f[i][j - v[i]] + w[i])

# print(f[n][m])

N = 1010

w = [0] * N
v = [0] * N
f = [0 for _ in range(N)]

n, m = map(int, input().split())

for i in range(1, n + 1):
    vi, wi = map(int, input().split())
    v[i] = vi
    w[i] = wi

for i in range(1, n + 1):
    for j in range(1, m + 1):
        if j >= v[i]:
            f[j] = max(f[j], f[j - v[i]] + w[i])

print(f[m])
```

338.计数问题

```
def count(num, x):
    s = str(num)
    n = len(s)

    res = 0
```

```
    for i in range(not x, n):
        if i > 0:
            res += int(s[:i]) * pow(10, n - i - 1)
            if x == 0:
                res -= pow(10, n - i - 1)
        if int(s[i]) == x:
            if s[i + 1:]:
                res += int(s[i + 1:])
            res += 1
        elif int(s[i]) > x:
            res += pow(10, n - i - 1)

    return res


while True:
    a, b = map(int, input().split())

    if a == 0 and b == 0:
        break
    if a < b:
        a, b = b, a

    for i in range(10):
        if i < 9:
            print(count(a, i) - count(b - 1, i), end=" ")
        else:
            print(count(a, i) - count(b - 1, i))
```

895.最长上升子序列

```
N = 1010

f = [0] * N

n = int(input())

arr = list(map(int, input().split()))

for i in range(n):
    f[i] = 1 # 重点，别记忘了，因为arr[i] 本身就是一个长度为 1 的上升子序列，如果前面的数字都比该数字大的话，没法上升，初始化为1
    for j in range(i):
        if arr[j] < arr[i]:
            f[i] = max(f[i], f[j] + 1)

print(max(f))
```

897.最长公共子序列

```
N = 1010


f = [[0 for _ in range(N)]for _ in range(N)]

n, m = map(int, input().split())
A = input()
B = input()

for i in range(n):
    for j in range(m):
        f[i][j] = max(f[i - 1][j], f[i][j - 1])
        if A[i] == B[j]:
            f[i][j] = max(f[i][j], f[i - 1][j - 1] + 1)

print(f[n - 1][m - 1])
```

898.数字三角形

```
N = 510

f = []

n = int(input())

for i in range(1, n + 1):
    f.append(list(map(int, input().split())))

for i in range(n - 2, -1, -1):
    for j in range(i + 1):
        f[i][j] = max(f[i + 1][j], f[i + 1][j + 1]) + f[i][j]

print(f[0][0])
```

899.编辑距离

```
def edit_distance(a, b):
    lena = len(a) - 1
    lenb = len(b) - 1
    f = [[0 for _ in range(lenb + 1)] for _ in range(lena + 1)]

    for i in range(lena + 1):
        f[i][0] = i
    for i in range(lenb + 1):
        f[0][i] = i
```

```
    for i in range(1, lena + 1):
        for j in range(1, lenb + 1):
            f[i][j] = min(f[i - 1][j] + 1, f[i][j - 1] + 1)
            if a[i] == b[j]:
                f[i][j] = min(f[i][j], f[i - 1][j - 1])
            else:
                f[i][j] = min(f[i][j], f[i - 1][j - 1] + 1)

    return f[lena][lenb]


n, m = map(int, input().split())

strs = []

for _ in range(n):
    strs.append(' ' + input())

for _ in range(m):
    c, limit = input().split()
    c = ' ' + c
    limit = int(limit)
    res = 0
    for i in range(n):
        if edit_distance(strs[i], c) <= limit:
            res += 1
    print(res)
```

900.整数划分

```
# N = 1010
# MOD = int(1e9) + 7

# f = [[0 for _ in range(N)] for _ in range(N)]

# f[1][1] = 1

# n = int(input())

# for i in range(2, n + 1):
#     for j in range(1, n + 1):
#         f[i][j] = (f[i - 1][j - 1] + f[i - j][j]) % MOD


# print(sum(f[n]) % MOD)
N = 1010
MOD = int(1e9) + 7

f = [[0 for _ in range(N)] for _ in range(N)]

n = int(input())

for i in range(1,n + 1):
```

```
    f[i][0] = 1

for i in range(1, n + 1):
    for j in range(1, n + 1):
        f[i][j] = f[i - 1][j] % MOD
        if j >= i:
            f[i][j] = (f[i][j] + f[i][j - i]) % MOD


print(f[n][n])
```

901.滑雪

```
N = 310

choices = [[-1, 0], [0, 1], [1, 0], [0, -1]]

g = []
st = [[-1 for _ in range(N)] for _ in range(N)]

r, c = map(int, input().split())

g.append([])
for _ in range(r):
    g.append([0] + list(map(int, input().split())))

print(g)


def check(i, j):
    global r, c
    if i <= r and i >= 1 and j <= c and j >= 1:
        return True
    else:
        return False


def dfs(i, j):
    global r, c
    if st[i][j] != -1:
        return st[i][j]
    res = 1
    for choice in choices:
        if check(i + choice[0], j + choice[1]) and g[i][j] > g[i +
choice[0]][j + choice[1]]:
            res = max(res, 1 + dfs(i + choice[0], j + choice[1]))
    st[i][j] = res
    return st[i][j]


def solution():
    global r,c
    max_depth = 1
```

```
    for i in range(1, r + 1):
        for j in range(1,c + 1):
            max_depth = max(max_depth,dfs(i,j))

    return max_depth

print(solution())
```

902.最短编辑距离

```
N = 1010

f = [[0 for _ in range(N)] for _ in range(N)]

n = int(input())

a = ' ' + input()

m = int(input())

b = ' ' + input()

for i in range(n + 1):
    f[i][0] = i
for i in range(m + 1):
    f[0][i] = i

for i in range(1, n + 1):
    for j in range(1, m + 1):
        f[i][j] = min(f[i - 1][j] + 1, f[i][j - 1] + 1)
        if a[i] == b[j]:
            f[i][j] = min(f[i][j], f[i - 1][j - 1])
        else:
            f[i][j] = min(f[i][j], f[i - 1][j - 1] + 1)

print(f[n][m])
```

91.最短Hamilton路径

```
N = 20
M = 1 << N

w = []
f = [[float("inf") for _ in range(N)] for _ in range(M)]
f[1][0] = 0

n = int(input())
```

```
for _ in range(n):
    w.append(list(map(int, input().split())))

for i in range(1 << n):
    for j in range(n):
        if (i >> j) & 1:
            for k in range(n):
                if ((i - (1 << j)) >> k) & 1:
                    f[i][j] = min(f[i][j], f[i - (1 << j)][k] + w[k][j])

print(f[(1 << n) - 1][n - 1])
```

第六讲

104.货仓选址

```
n = int(input())

arr = list(map(int,input().split()))

arr.sort()

res = 0
for i in range(n):
    res +=  arr[n - i - 1] - arr[i]

print(res)
```

125.耍杂技的牛

```
a = []

n = int(input())

for _ in range(n):
    a.append(list(map(int,input().split())))

a.sort(key=lambda x:x[0] + x[1])

N = 50000 + 10
s = [0] * N


res = float("-inf")
SUM = 0
for i in range(n):
```

```
    w = a[i][0]
    s = a[i][1]
    res = max(res,SUM - s)
    SUM += w
print(res)
```

148.合并果子

```
import heapq

N = int(1e5) + 10

arr = []

n = int(input())

arr = list(map(int, input().split()))


heapq.heapify(arr)

res = 0
now = 0
while len(arr) > 1:
    a = heapq.heappop(arr)
    b = heapq.heappop(arr)
    res += a + b

    heapq.heappush(arr, a + b)

print(res)
```

905.区间选点

```
N = int(1e5) + 10

a = []

n = int(input())

for _ in range(n):
    a.append(list(map(int, input().split())))

a.sort(key=lambda x: x[1])

res = 0
ed = -2e9
for i in range(n):
    if a[i][0] > ed:
```

```
        res += 1
        ed = a[i][1]

print(res)
```

906.区间分组

```
import heapq

heap = []
arr = []

n = int(input())

for _ in range(n):
    arr.append(list(map(int, input().split())))

arr.sort(key=lambda x: x[0])



for i in range(n):
    if len(heap) == 0 or heap[0] >= arr[i][0]:
        heapq.heappush(heap, arr[i][1])
    else:
        heapq.heappop(heap)
        heapq.heappush(heap, arr[i][1])

print(len(heap))
```

907.区间覆盖

```
# 暴力写法
# N = int(1e5) + 10

# arr = []

# s, t = map(int, input().split())

# n = int(input())

# for _ in range(n):
#     arr.append(list(map(int, input().split())))

# arr.sort(key=lambda x: x[0])
# res = 0
# while s <= t:
#     max_r = -2e9
```

```
#     for j in range(n):
#         if arr[j][0] <= s:
#             max_r = max(max_r, arr[j][1])
#         else:
#             break

#     if max_r == -2e9:
#         res = - 1
#         break
#     res += 1
#     s = max_r


# print(res)

arr = []

s, t = map(int, input().split())

n = int(input())

for _ in range(n):
    arr.append(list(map(int, input().split())))

arr.sort(key=lambda x: x[0])

res = 0
suscess = False
i  =  0
while i < n:
    j = i
    r = -2e9

    while j < n and arr[j][0] <= s:
        r = max(r,arr[j][1])
        j += 1

    if r < s:
        res = -1
        break

    res += 1
    if r >= t:
        suscess = True
        break

    s = r
    i = j

print(res if suscess else -1)
```

908.最大不相交区间数量

```
N = int(1e5) + 10

a = []

n = int(input())

for _ in range(n):
    a.append(list(map(int, input().split())))

a.sort(key=lambda x: x[1])

res = 0
ed = -2e9
for i in range(len(a)):
    if ed < a[i][0]:
        res += 1
        ed = a[i][1]

print(res)
```

913.排队打水

```
N = int(1e5) + 10

arr = []

n = int(input())

arr = list(map(int, input().split()))

arr.sort()

res = 0

for i in range(n):
    res += arr[i] * (n - i - 1)

print(res)
```

第四讲

204.表达整数的奇怪方式

```
k1, k2 = 0, 0


def exgcd(a, b):
    global k1, k2

    if b == 0:
        k1 = 1
        k2 = 0
        return a
    else:
        d = exgcd(b, a % b)
        tmp = k1
        k1 = k2
        k2 = tmp - a // b * k2
        return d

x = 0
n = int(input())
m1, a1 = map(int, input().split())
for _ in range(n - 1):
    m2, a2 = map(int, input().split())
    d = exgcd(m1, m2)
    if (a2 - a1) % d:
        x = -1
        break
    k1 *= (a2 - a1) // d
    k1 = k1 % (m2 // d)

    x = k1 * m1 + a1

    a1 = x
    m1 = m1 // d * m2

if x != -1:
    x = a1 % m1
print(x)
```

866.试除法判定质数

```
import math

n = int(input())

def is_prime(n):
    if n < 2:
        return False
    for i in range(2,int(math.sqrt(n)) + 1):
        if n % i == 0:
            return False
```

```
for i in range(n):
    num = int(input())
    if is_prime(num) == False:
        print("Yes")
    else:
        print("No")
```

868.筛质数

```
N = int(1e6 + 10)
primes = [0] * N
st = [0] * N
cnt = 0

n = int(input())

for i in range(2, n + 1):
    if st[i] == 0:
        primes[cnt] = i
        cnt += 1
    j = 0
    while primes[j] <= n // i:
        st[primes[j] * i] = 1
        if i % primes[j] == 0:
            break
        j += 1

print(cnt)
```

869.试除法求约数

```
n = int(input())

def num(a):
    arr = []
    i = 1
    for i in range(1,(a // i) + 1):
        if a % i == 0:
            arr.append(i)
            if i != a // i:
                arr.append(a // i)
    arr.sort()
    print(" ".join(map(str,arr)))

for _ in range(n):
    a = int(input())
    num(a)
```

```
# def divide(n):
#     i = 1
#     f = []
#     while i <= n // i:
#         if n % i == 0:
#             f.append(i)
#             if i != n // i:
#                 f.append(n//i)

#         i += 1
#     f.sort()
#     return f

# n = int(input())
# while n:
#     n -= 1
#     a = int(input())
#     t = divide(a)
#     for i in t:
#         print(i,end=" ")
#     print()
```

870.约数个数

```
from collections import defaultdict

N = 110
MOD = int(1e9) + 7
primes = defaultdict(int)

n = int(input())

for _ in range(n):
    a = int(input())
    i = 2

    while i * i <= a:

        while a % i == 0:
            a //= i
            primes[i] += 1
        i += 1
    if a > 1:  # 如果a本身是个质数
        primes[a] += 1

res = 1
for i in primes.values():
    res = int(res * (i + 1) % MOD)
print(res)
```

871.约数之和

```
from collections import defaultdict
MOD = int(1e9) + 7
primes = defaultdict(int)

n = int(input())

for _ in range(n):
    a = int(input())

    i = 2
    while i * i <= a:
        while a % i == 0:
            a //= i
            primes[i] += 1
        i += 1
    if a > 1:
        primes[a] += 1

res = 1


for key, value in primes.items():
    t = 1
    for _ in range(1, value + 1):
        t = (t * key + 1) % MOD
    res = res * t % MOD

print(res)
```

872.最大公约数

```
# from math import gcd

# n = int(input())

# for _ in range(n):
#     a, b = map(int, input().split())
#     print(gcd(a,b))


def gcd(a,b):
    return gcd(b,a % b) if b != 0 else a

n = int(input())

for _ in range(n):
    a, b = map(int, input().split())
    print(gcd(a,b))
```

873.欧拉函数

```
n = int(input())

for _ in range(n):
    a = int(input())

    i = 2

    res = a
    while i * i <= a:

        if a % i == 0:
            # 注意这里，i会比1小很多，如果用//这里是0，则整个式子为res;或者写成 res = res / i  * (i - 1 ),因为res一定能整除i
            res = int(res * (1 - 1 / i))
            while a % i == 0:
                a //= i
        i += 1
    if a > 1:
        res = int(res * (1 - 1 / a))
    print(res)
```

874.筛法求欧拉函数

```
N = int(1e6) + 10

phi = [0] * N
primes = [0] * N
st = [0] * N
cnt = 0

n = int(input())

phi[1] = 1
for i in range(2, n+1):
    if st[i] == 0:
        primes[cnt] = i
        cnt += 1
        phi[i] = i - 1
    j = 0

    while primes[j] * i <= n:
        st[primes[j] * i] = 1
        if i % primes[j] == 0:
            phi[primes[j] * i] = primes[j] * phi[i]
            break
        else:
            phi[primes[j] * i] = phi[i] * (primes[j] - 1)
```

```
        j += 1

print(sum(phi[1:n + 1]))
```

875.快速幂

```
def qmi(a,b,p):
    res = 1
    while b:
        if b & 1:
            res = res * a % p
        b >>= 1
        a *= a % p
    return res

n =int(input())

for _ in range(n):
    a,b,p = map(int,input().split())
    print(qmi(a,b,p))
```

876.快速幂求逆元

```
def qmi(a, b, p):
    res = 1
    while b:
        if b & 1:
            res = res * a % p
        b >>= 1
        a = a * a % p
    return res


n = int(input())

for _ in range(n):
    a, p = map(int, input().split())

    if a % p:
        print(qmi(a, p-2, p))
    else:
        print("impossible")
```

877.扩展欧几里得算法

```python
def exgcd(a, b):
    global x, y
    if b:
        d = exgcd(b, a % b)
        tmp = y
        y = x - (a // b) * y
        x = tmp
        return d
    else:
        x = 1
        y = 0
        return a


n = int(input())

x, y = 0, 0

for _ in range(n):
    a, b = map(int, input().split())
    exgcd(a, b)
    print(x, y)
```

878.线性同余方程

```python
x, y = 0, 0


def exgcd(a, b):
    global x, y
    if b == 0:
        x = 1
        y = 0
        return a
    d = exgcd(b, a % b)
    tmp = x
    x = y
    y = tmp - a // b * y

    return d


n = int(input())

for _ in range(n):
    a, b, m = map(int, input().split())
    d = exgcd(a, m)

    if b % d:
        print("impossible")
    else:
```

```
        print(b // d * x % m)
```

884.高斯消元解异或线性方程组

```
arr = []


def gauss():
    r = 0

    for c in range(n):

        t = r

        for i in range(r, n):
            if arr[i][c]:
                t = i
        if arr[t][c] == 0:
            continue

        arr[r], arr[t] = arr[t], arr[r]

        for i in range(r + 1, n):
            if arr[i][c]:
                for j in range(n, c - 1, -1):
                    arr[i][j] = arr[i][j] ^arr[r][j]
        r += 1

    if r < n:
        for i in range(r, n):
            if arr[i][n] == 1:
                return 0
        return 2

    for i in range(n - 1, -1, -1):
        for j in range(i + 1, n):
            arr[i][n] = arr[i][n] ^ (arr[i][j] * arr[j][n])

    return 1


n = int(input())

for _ in range(n):
    arr.append(list(map(int, input().split())))

res = gauss()

if res == 0:
    print("No solution")
elif res == 2:
    print("Multiple sets of solutions")
elif res == 1:
```

```
    for i in range(n):
        print(arr[i][n])
```

885.求组合数I

```
N = 2000 + 10
MOD = int(1e9) + 7

c = [[0 for _ in range(N)] for _ in range(N)]

def init():
    for i in range(N):
        for j in range(i + 1):
            if j == 0:
                c[i][j] = 1
            else:
                c[i][j] = (c[i - 1][j] + c[i - 1][j - 1]) % MOD

n = int(input())

init()

for _ in range(n):
    a,b = map(int,input().split())
    print(c[a][b])
```

886.求组合数II

```
N = int(1e5) + 10
MOD = int(1e9) + 7

fact = [0] * N
infact = [0] * N


def qmi(a, b, p):
    res = 1
    while b:
        if b & 1:
            res *= a % p
        b >>= 1
        a *= a % p
    return res


fact[0], infact[0] = 1, 1
for i in range(1, N):
    fact[i] = (fact[i - 1] * i) % MOD
    infact[i] = (infact[i - 1] * (qmi(i, MOD - 2, MOD))) % MOD
```

```
n = int(input())

for _ in range(n):
    a, b = map(int, input().split())
    print((fact[a] * infact[b] * infact[a - b]) % MOD)
```

887.求组合数III

```
def qmi(a, b, p):
    res = 1
    while b:
        if b & 1:
            res = res * a % p  # 注意这里不能写成res *= a % p，因为这样的话会先算a
% p再 * res
        b >>= 1
        a = a * a % p
    return res


def C(a, b, p):
    if b > a:
        return 0
    res = 1
    j = a
    i = 1
    while i <= b:
        res = res * j % p
        res = res * qmi(i, p - 2, p) % p
        i += 1
        j -= 1
    return res


def lucas(a, b, p):
    if a < p and b < p:
        return C(a, b, p)
    return C(a % p, b % p, p) * lucas(a // p, b // p, p) % p


n = int(input())

for _ in range(n):
    a, b, p = map(int, input().split())
    print(lucas(a, b, p))
```

888.求组合数IV

```python
# import math
# n, m = map(int, input().split())
# print(math.factorial(n) // math.factorial(n - m) // math.factorial(m))

N = 5000 + 10
primes = [0] * N
cnt = 0
st = [0] * N
SUM = [0] * N


def get_primes(n):
    global cnt
    for i in range(2, n + 1):
        if st[i] == 0:
            st[i] = 1
            primes[cnt] = i
            cnt += 1
        j = 0
        while primes[j] * i <= n:
            st[primes[j] * i] = 1
            if i % primes[j] == 0:
                break
            j += 1


def get(n, p):
    res = 0
    while n:
        res += n // p
        n //= p
    return res

def mul(a,b):
    c = []
    t = 0
    for i in range(len(a)):
        t += a[i] * b
        c.append(t % 10)
        t //= 10
    while t:
        c.append(t % 10)
        t //= 10
    return c

a, b = map(int, input().split())

get_primes(a)

for i in range(cnt):
    prime = primes[i]
    SUM[i] = get(a, prime) - get(b, prime) - get(a - b, prime)

res = []
res.append(1)
for i in range(cnt):
```

```
    for j in range(SUM[i]):
        res = mul(res,primes[i])

for i in range(len(res) - 1,- 1, -1):
    print(res[i],end="")
```

889.满足条件的01序列

```
MOD = int(1e9) + 7  # 卡特兰数


def qmi(a, b, p):
    res = 1
    while b:
        if b & 1:
            res = res * a % p
        b >>= 1
        a = a * a % p
    return res

n = int(input())

a, b = 2*n, n
res = 1

for i in range(a, a-b, -1):
    res = res * i % MOD

for i in range(1, b + 1):
    res = res * qmi(i, MOD - 2, MOD) % MOD

res = res * qmi(n + 1, MOD - 2, MOD) % MOD

print(res)
```

890.能被整除的数

```
p = []
n,m = map(int,input().split())

p = list(map(int,input().split()))

res = 0
for i in range(1,1 << m):
    t = 1
    cnt = 0
    for j in range(m):
```

```
        if i >> j & 1:
            if p[j] * t > n:
                t = -1
                break
            cnt += 1
            t = t * p[j]
    if t != -1:
        if cnt % 2:
            res += n // t
        else:
            res -= n // t

print(res)
```

892.台阶-Nim游戏

```
res = 0
n = int(input())

arr = list(map(int,input().split()))

for i in range(n):
    if i % 2 == 0:
        res ^= arr[i]

if res:
    print("Yes")
else:
    print("No")
```

893.集合-Nim游戏

```
N = 110
M = 10010
f = [-1] * M


def sg(x):
    S = set()
    global m
    if f[x] != -1:
        return f[x]
    for i in range(m):
        if x >= s[i]:
            S.add(sg(x - s[i]))
    return mex(S, x)
```

```
def mex(S, x):
    i = 0
    while True:
        if i not in S:
            f[x] = i
            return i
        i += 1


m = int(input())

s = list(map(int, input().split()))

n = int(input())

h = list(map(int, input().split()))

res = 0
for i in range(n):
    res ^= sg(h[i])

print("Yes" if res != 0 else "No")
```

894.拆分-Nim游戏

```
N = 110
f = [-1] * N


def sg(x):
    global n
    S = set()
    if f[x] != -1:
        return f[x]
    for i in range(x):
        for j in range(i + 1):
            S.add(sg(i) ^ sg(j))
    return mex(S, x)


def mex(S, x):
    i = 0
    while True:
        if i not in S:
            f[x] = i
            return i
        i += 1


n = int(input())
a = list(map(int, input().split()))

res = 0
```

```python
for i in range(n):
    res ^= sg(a[i])

print("Yes" if res != 0 else "No")
```

算法提高课

第一章动态规划

10.有依赖的背包问题

```cpp
#include <iostream>
#include <algorithm>
#include <cstring>

using namespace std;

const int N = 110;

int f[N][N];
int e[N], ne[N], h[N], idx;
int v[N], w[N];
int root;
int n, m;

void add(int a, int b)
{
    e[idx] = b;
    ne[idx] = h[a];
    h[a] = idx++;
}

void dfs(int u)
{
    for (int i = v[u]; i <= m; i++)
    {
        f[u][i] = w[u];
    }

    for (int i = h[u]; i != -1; i = ne[i])
    {
        int son = e[i];
        dfs(son);

        for (int j = m; j >= v[u]; j--)
        {
            for (int k = 0; k <= j - v[u]; k++)
```

```
            {
                f[u][j] = max(f[u][j], f[u][j - k] + f[son][k]);
            }
        }
    }
}

int main()
{
    memset(h, -1, sizeof(h));
    cin >> n >> m;
    for (int i = 1; i <= n; i++)
    {
        int p;
        cin >> v[i] >> w[i] >> p;
        if (p == -1)
        {
            root = i;
        }
        else
        {
            add(p, i);
        }
    }
    dfs(root);

    cout << f[root][m] << endl;
    return 0;
}
```

```
N = 110

e = [0] * N
ne = [0] * N
h = [-1] * N
v = [0] * N
w = [0] * N
f = [[0 for _ in range(N)]for _ in range(N)]
idx = 0
root = -1


def add(a, b):
    global idx
    e[idx] = b
    ne[idx] = h[a]
    h[a] = idx
    idx += 1


def dfs(u):
    global m
    for i in range(v[u], m + 1):
        f[u][i] = w[u]
```

```
    i = h[u]
    while i != -1:
        son = e[i]
        dfs(son)
        for j in range(m, v[u] - 1, -1):
            for k in range(0, j - v[u] + 1):
                f[u][j] = max(f[u][j], f[u][j - k] + f[son][k])

        i = ne[i]


n, m = map(int, input().split())

for i in range(1, n + 1):
    v[i], w[i], p = map(int, input().split())
    if p == -1:
        root = i
    else:
        add(p, i)

dfs(root)

print(f[root][m])
```

1010.拦截导弹

```
#include <iostream>
#include <cstring>
#include <algorithm>

using namespace std;

const int N = 1010;

int f[N], a[N], g[N];

int main()
{

    int n = 0;
    while (cin >> a[++n])
        ;
    n--;
    int res = 0;
    for (int i = 1; i <= n; i++)
    {
        f[i] = 1;
        for (int j = 1; j < i; j++)
        {
            if (a[i] <= a[j])
                f[i] = max(f[i], f[j] + 1);
        }
        res = max(res, f[i]);
```

```
    }
    printf("%d\n", res);

    int cnt = 0;
    for (int i = 1; i <= n; i++)
    {
        int k = 0;
        while (k < cnt && g[k] < a[i])
        {
            k++;
        }
        g[k] = a[i];
        if (k >= cnt)
            cnt++;
    }
    printf("%d\n", cnt);

    return 0;
}
```

```
N = 1010

f = [0] * N
g = [0] * N

a = []


a = [0] + list(map(int, input().split()))

n = len(a) - 1

cnt = 0


for i in range(n, 0, -1):
    f[i] = 1
    for j in range(n, i, -1):
        if a[i] >= a[j]:
            f[i] = max(f[i], f[j] + 1)
    cnt = max(cnt, f[i])

print(cnt)

res = 0

for i in range(1, n + 1):
    k = 0
    while k < res and g[k] < a[i]:
        k += 1
    g[k] = a[i]
    if k >= res:
        res += 1
```

```
print(res)
```

1012.友好城市

```
#include <iostream>
#include <cstring>
#include <algorithm>

using namespace std;

typedef pair<int, int> PII;

const int N = 5010;

int f[N];
PII a[N];

int main()
{
    int n;
    cin >> n;
    for (int i = 0; i < n; i++)
        cin >> a[i].first >> a[i].second;

    sort(a, a + n);

    int res = 0;
    for (int i = 0; i < n; i++)
    {
        f[i] = 1;
        for (int j = 0; j < i; j++)
        {
            if (a[i].second > a[j].second)
            {
                f[i] = max(f[i], f[j] + 1);
                res = max(res, f[i]);
            }
        }
    }
    printf("%d\n", res);

    return 0;
}
```

```
N = 5000 + 10

f = [0] * N
a = []

n = int(input())
```

```python
for _ in range(n):
    a.append(list(map(int, input().split())))

a.sort(key=lambda x: x[0])

res = 0
for i in range(n):
    f[i] = 1
    for j in range(i):
        if a[i][1] > a[j][1]:
            f[i] = max(f[i], f[j] + 1)
            res = max(res, f[i])
print(res)
```

1013.机器分配

```cpp
#include <iostream>
#include <algorithm>
#include <cstring>

using namespace std;

const int N = 15, M = 20;

int f[N][M];
int a[N][M];
int way[N];

int main()
{
    int n, m;
    cin >> n >> m;
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= m; j++)
        {
            cin >> a[i][j];
        }
    }
    for (int i = 1; i <= n; i++)
    {
        for (int j = 0; j <= m; j++)
        {
            for (int k = 0; k <= j; k++)
            {
                f[i][j] = max(f[i][j], f[i - 1][j - k] + a[i][k]);
            }
        }
    }
    cout << f[n][m] << endl;
    int j = m;
    for (int i = n; i > 0; i--)
```

```
    {
        for (int k = 1; k <= j; k++)
        {
            if (f[i][j] == f[i - 1][j - k] + a[i][k])
            {
                way[i] = k;
                j -= k;
                break;
            }
        }
    }
    for (int i = 1; i <= n; i++)
        cout << i << " " << way[i] << endl;
    return 0;
}
```

```
N = 20
v = [0] * N
w = [0] * N
f = [[0 for _ in range(N)] for _ in range(N)]
res = [0] * N
n, m = map(int, input().split())

a = []

for i in range(n):
    a.append([0] + list(map(int, input().split())))

for i in range(0, n):
    for j in range(m, -1, -1):
        f[i][j] = f[i - 1][j]
        for k in range(1, m + 1):
            if j >= k:
                f[i][j] = max(f[i][j], f[i - 1][j - k] + a[i][k])

print(f[n - 1][m])

j = m

way = [0] * N
for i in range(n - 1, -1, -1):
    for k in range(0, j + 1):
        if f[i][j] == f[i - 1][j - k] + a[i][k]:
            way[i] = k
            j -= k
            break

for i in range(n):
    print(i+ 1,way[i])
```

1014.登山

```cpp
#include <iostream>
#include <cstring>
#include <algorithm>

using namespace std;

const int N = 1010;
int f[N], g[N];
int a[N];

int main()
{
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++)
        cin >> a[i];
    for (int i = 1; i <= n; i++)
    {
        f[i] = 1;
        for (int j = 1; j < i; j++)
        {
            if (a[i] > a[j])
                f[i] = max(f[i], f[j] + 1);
        }
    }
    for (int i = n; i >= 1; i--)
    {
        g[i] = 1;
        for (int j = n; j > i; j--)
        {
            if (a[i] > a[j])
                g[i] = max(g[i], g[j] + 1);
        }
    }
    int res = 0;
    for (int i = i; i <= n; i++)
        res = max(res, f[i] + g[i] - 1);
    printf("%d\n", res);
    return 0;
}
```

```python
N = 1010

f = [0] * N
g = [0] * N

n = int(input())

a = [0] + list(map(int, input().split()))

for i in range(1, n + 1):
    f[i] = 1
```

```
    for j in range(1, i):
        if a[i] > a[j]:
            f[i] = max(f[i], f[j] + 1)

for i in range(n, 0, -1):
    g[i] = 1
    for j in range(n, i, -1):
        if a[i] > a[j]:
            g[i] = max(g[i], g[j] + 1)

res = 0
for k in range(1, n + 1):
    res = max(res, f[k] + g[k] - 1)

print(res)
```

1015.摘花生

```
#include <iostream>
#include <cstring>
#include <algorithm>

using namespace std;

const int N = 110;

int w[N][N], f[N][N];

int main()
{
    int T;
    cin >> T;
    while (T--)
    {
        int r, c;
        cin >> r >> c;
        for (int i = 1; i <= r; i++)
        {
            for (int j = 1; j <= c; j++)
            {
                cin >> w[i][j];
            }
        }

        for (int i = 1; i <= r; i++)
        {
            for (int j = 1; j <= c; j++)
            {
                f[i][j] = max(f[i - 1][j], f[i][j - 1]) + w[i][j];
            }
        }
        printf("%d\n", f[r][c]);
```

```
    }
}
```

```
N = 110

f = [[0 for _ in range(N)]for _ in range(N)]

t = int(input())

for _ in range(t):
    a = []
    f = [[0 for _ in range(N)]for _ in range(N)]
    r, c = map(int, input().split())
    for _ in range(r):
        a.append(list(map(int, input().split())))
    for i in range(1, r + 1):
        for j in range(1, c + 1):
            f[i][j] = max(f[i - 1][j] + a[i - 1][j - 1], f[i][j])
            f[i][j] = max(f[i][j - 1] + a[i - 1][j - 1], f[i][j])
    print(f[r][c])
```

1016.最大上升子序列和

```
N = 1010

a = []

f = [0] * N

n = int(input())

a = [0] + list(map(int, input().split()))


for i in range(1, n + 1):
    f[i] = a[i]
    for j in range(1, i):
        if a[i] > a[j]:
            f[i] = max(f[i], f[j] + a[i])


print(max(f))
```

```
#include <iostream>
#include <cstring>
#include <algorithm>
```

```cpp
using namespace std;

const int N = 1010;

int f[N], a[N];

int main()
{
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++)
        cin >> a[i];
    int res = 0;
    for (int i = 1; i <= n; i++)
    {
        f[i] = a[i];
        for (int j = 1; j < i; j++)
        {
            if (a[i] > a[j])
                f[i] = max(f[i], f[j] + a[i]);
        }
        res = max(res, f[i]);
    }
    printf("%d\n", res);
    return 0;
}
```

1017.怪盗基德的滑翔翼

```cpp
#include <iostream>
#include <cstring>
#include <algorithm>

using namespace std;

const int N = 55;
int f1[N];
int f2[N];
int h[N];

int main()
{
    int k;
    cin >> k;
    while (k--)
    {
        int n;
        cin >> n;
        for (int i = 1; i <= n; i++)
            cin >> h[i];
        for (int i = 1; i <= n; i++)
        {
            f1[i] = 1;
```

```
            f2[i] = 1;
            for (int j = 1; j < i; j++)
            {
                if (h[i] > h[j])
                {
                    f1[i] = max(f1[i], f1[j] + 1);
                }
                if (h[i] < h[j])
                {
                    f2[i] = max(f2[i], f2[j] + 1);
                }
            }
        }
        printf("%d\n", max(f1[n], f2[n]));
    }

    return 0;
}
```

```
# 多循环，初始化

N = 110

f1 = [0] * N
f2 = [0] * N
h = []

k = int(input())

for _ in range(k):
    f1 = [0] * N
    f2 = [0] * N
    n = int(input())
    h = list(map(int,input().split()))

    for i in range(n):
        f1[i] = 1
        f2[i] = 1
        for j in range(i):
            if h[i] > h[j]:
                f1[i] = max(f1[i],f1[j] + 1)
            if h[i] < h[j]:
                f2[i] = max(f2[i],f2[j] + 1)

    print(max(max(f1),max(f2)))
```

1018.最低通行费

```
#include <iostream>
#include <algorithm>
#include <cstring>
```

```cpp
using namespace std;

const int N = 110, INF = 1e9;

int f[N][N];
int w[N][N];

int main()
{
    int n;
    scanf("%d", &n);
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= n; j++)
        {
            scanf("%d", &w[i][j]);
        }
    }
    // memset(f, INF, sizeof(f));
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= n; j++)
        {
            if (i == 1 && j == 1)
            {
                f[i][j] = w[i][j];
            }
            else
            {
                f[i][j] = INF;
                if (i > 1)
                    f[i][j] = min(f[i][j], f[i - 1][j] + w[i][j]);
                if (j > 1)
                    f[i][j] = min(f[i][j], f[i][j - 1] + w[i][j]);
            }
        }
    }
    printf("%d\n", f[n][n]);
}
```

```python
N = 110

f = [[float("inf") for _ in range(N)] for _ in range(N)]

w = []

n = int(input())

for _ in range(n):
    w.append(list(map(int, input().split())))


for i in range(1, n + 1):
    for j in range(1, n + 1):
```

```
        if i == 1 and j == 1:
            f[i][j] = w[i - 1][j - 1]
        if i > 1:
            f[i][j] = min(f[i][j], f[i - 1][j] + w[i - 1][j - 1])
        if j > 1:
            f[i][j] = min(f[i][j], f[i][j - 1] + w[i - 1][j - 1])

print(f[n][n])
```

1019.庆功会

```
#include <iostream>
#include <algorithm>
#include <cstring>

using namespace std;

const int N = 6000 + 10;

int f[N];

int main()
{
    int n, m;
    cin >> n >> m;
    for (int i = 0; i < n; i++)
    {
        int v, w, s;
        cin >> v >> w >> s;
        for (int j = m; j > 0; j--)
        {
            for (int k = 1; k <= s && j >= k * v; k++)
            {
                f[j] = max(f[j], f[j - k * v] + k * w);
            }
        }
    }
    cout << f[m] << endl;
    return 0;
}
```

```
N = 6000 + 10

f = [0] * N

n, m = map(int, input().split())

for i in range(n):
    v, w, s = map(int, input().split())
    for j in range(m, 0, -1):
```

```
        k = 1
        while k <= s and j >= k * v:
            f[j] = max(f[j], f[j - k * v] + k * w)
            k += 1

print(f[m])
```

1020.潜水员

```
#include <iostream>
#include <algorithm>
#include <cstring>

using namespace std;

const int M = 25, N = 85;
int f[M][N];

int main()
{
    int m, n;
    cin >> m >> n;
    int k;
    cin >> k;
    memset(f, 0x3f, sizeof(f));
    f[0][0] = 0;
    while (k--)
    {
        int a, b, c;
        cin >> a >> b >> c;
        for (int i = m; i >= 0; i--)
        {
            for (int j = n; j >= 0; j--)
            {
                f[i][j] = min(f[i][j], f[max(0, i - a)][max(0, j - b)] + c);
            }
        }
    }
    cout << f[m][n] << endl;

    return 0;
}
```

```
M = 25
N = 85

f = [[float("inf") for _ in range(N)] for _ in range(N)]
f[0][0] = 0

m, n = map(int, input().split())
```

```
k = int(input())

for _ in range(k):
    a, b, c = map(int, input().split())
    for i in range(m, -1, -1):
        for j in range(n, -1, -1):
            f[i][j] = min(f[i][j], f[max(0, i - a)][max(0, j - b)] + c)

print(f[m][n])
```

1021.货币系统

```
#include <iostream>
#include <algorithm>
#include <cstring>

using namespace std;

typedef long long LL;

const int N = 16, M = 3000 + 5;

LL f[M];

LL a[N];

int main()
{
    int n, m;
    cin >> n >> m;
    f[0] = 1;
    for (int i = 1; i <= n; i++)
    {
        int w;
        cin >> w;
        for (int j = 1; j <= m; j++)
        {
            if (j >= w)
                f[j] += f[j - w];
        }
    }
    cout << f[m];
    return 0;
}
```

```
N = 16
M = 3000 + 5

f = [[0 for _ in range(M)] for _ in range(N)]
```

```python
a = [0] * N

n, m = map(int, input().split())

for i in range(1, n + 1):
    a[i] = int(input())

for i in range(1, n + 1):
    f[i][0] = 1
    for j in range(1, m + 1):
        f[i][j] = f[i - 1][j]
        if j >= a[i]:
            f[i][j] += f[i][j - a[i]]

print(f[n][m])
```

1022.宠物小精灵之收服

```cpp
#include <iostream>
#include <algorithm>
#include <cstring>

using namespace std;

const int N = 1010, M = 510, K = 105;

int f[N][M];

int main()
{
    int n, m, k;
    cin >> n >> m >> k;
    for (int i = 1; i <= k; i++)
    {
        int v1, v2;
        cin >> v1 >> v2;
        for (int j = n; j >= v1; j--)
        {
            for (int z = m - 1; z >= v2; z--)
            {
                f[j][z] = max(f[j][z], f[j - v1][z - v2] + 1);
            }
        }
    }
    cout << f[n][m - 1] << " ";
    int z = m - 1;
    while (z > 0 && f[n][z - 1] == f[n][m - 1])
        z--;
    cout << m - z << endl;
    return 0;
}
```

```python
N = 1010
M = 510
K = 105

f = [[0 for _ in range(M)]for _ in range(N)]

n, m, k = map(int, input().split())

for i in range(1, k + 1):
    v1, v2 = map(int, input().split())
    for j in range(n, v1 - 1, -1):
        for z in range(m - 1, v2 - 1, -1):
            f[j][z] = max(f[j][z], f[j - v1][z - v2] + 1)

k = m - 1

while k > 0 and f[n][k - 1] == f[n][m - 1]:
    k -= 1

print(f[n][m - 1],m - k)
```

1023.买书

```cpp
#include <iostream>
#include <algorithm>
#include <cstring>

using namespace std;

const int N = 1010;

int f[N];
int a[] = {0, 10, 20, 50, 100};

int main()
{
    int n;
    cin >> n;
    f[0] = 1;
    for (int i = 1; i <= 4; i++)
    {
        for (int j = i; j <= n; j++)
        {
            if (j >= a[i])
                f[j] += f[j - a[i]];
        }
    }

    cout << f[n];
```

```
    return 0;
}
```

```
N = 1010

f = [[0 for _ in range(N)]for _ in range(5)]

a = [0, 10, 20, 50, 100]

n = int(input())


for i in range(1, 5):
    f[i][0] = 1
    for j in range(1, n + 1):
        f[i][j] = f[i - 1][j]
        if j >= a[i]:
            f[i][j] += f[i][j - a[i]]

print(f[4][n])
```

1024.装箱问题

```
#include <algorithm>
#include <iostream>
#include <cstring>

using namespace std;

const int N = 35, V = 20000 + 10;

int f[N][V];
int a[N];

int main()
{
    int v, n;
    cin >> v >> n;
    for (int i = 1; i <= n; i++)
    {
        cin >> a[i];
        for (int j = 1; j <= v; j++)
        {
            f[i][j] = f[i - 1][j];
            if (j >= a[i])
            {
                f[i][j] = max(f[i][j], f[i - 1][j - a[i]] + a[i]);
            }
        }
    }
```

```
    cout << (v - f[n][v]) << endl;
    return 0;
}
```

```
V = 20000 + 10
N = 35

f = [[0 for _ in range(V)]for _ in range(N)]

a = []

v = int(input())
n = int(input())

a.append(0)
for _ in range(n):
    a.append(int(input()))

for i in range(1, n + 1):
    for j in range(1, v + 1):
        f[i][j] = f[i - 1][j]
        if j >= a[i]:
            f[i][j] = max(f[i][j], f[i - 1][j - a[i]] + a[i])

print(v - f[n][v])
```

1027.方格取数

```
#include <iostream>
#include <cstring>
#include <algorithm>

using namespace std;

const int N = 11;

int f[2 * N][N][N];
int w[N][N];

int main()
{
    int n;
    cin >> n;
    int a, b, c;
    while (cin >> a >> b >> c, a || b || c)
    {
        w[a][b] = c;
    }
    for (int k = 2; k <= 2 * n; k++)
    {
```

```
        for (int i1 = 1; i1 <= n; i1++)
        {
            for (int i2 = 1; i2 <= n; i2++)
            {
                int j1 = k - i1, j2 = k - i2;
                if (j1 >= 1 && j1 <= n && j2 >= 1 && j2 <= n)
                {
                    int t = w[i1][j1];
                    if (i1 != i2)
                        t += w[i2][j2];
                    f[k][i1][i2] = max(f[k][i1][i2], f[k - 1][i1 - 1][i2 -
1] + t);
                    f[k][i1][i2] = max(f[k][i1][i2], f[k - 1][i1 - 1][i2] +
t);
                    f[k][i1][i2] = max(f[k][i1][i2], f[k - 1][i1][i2 - 1] +
t);
                    f[k][i1][i2] = max(f[k][i1][i2], f[k - 1][i1][i2] + t);
                }
            }
        }
    }
    printf("%d\n", f[2 * n][n][n]);
    return 0;
}
```

```
N = 11

f = [[[0 for _ in range(N)]for _ in range(N)] for _ in range(2 * N)]
w = [[0 for _ in range(N)]for _ in range(N)]
n = int(input())

while True:
    a, b, c = map(int, input().split())
    if a == 0 and b == 0 and c == 0:
        break
    w[a][b] = c

for k in range(2, 2 * n + 1):
    for i1 in range(1, n + 1):
        for i2 in range(1, n + 1):
            j1 = k - i1
            j2 = k - i2
            if j1 >= 1 and j1 <= n and j2 >= 1 and j2 <= n:
                t = w[i1][j1]
                if i1 != i2:
                    t += w[i2][j2]
                f[k][i1][i2] = max(f[k][i1][i2], f[k - 1][i1 - 1][i2 - 1] +
t)
                f[k][i1][i2] = max(f[k][i1][i2], f[k - 1][i1 - 1][i2] + t)
                f[k][i1][i2] = max(f[k][i1][i2], f[k - 1][i1][i2 - 1] + t)
                f[k][i1][i2] = max(f[k][i1][i2], f[k - 1][i1][i2] + t)

print(f[2 * n][n][n])
```

1049.大盗阿福

```cpp
#include <algorithm>
#include <iostream>
#include <cstring>

using namespace std;

const int N = 1e5 + 10, INF = 0x3f3f3f3f;

int f[N][2];
int a[N];

int main()
{
    int t;
    cin >> t;
    while (t--)
    {
        int n;
        cin >> n;
        for (int i = 1; i <= n; i++)
            cin >> a[i];
        f[0][0] = 0, f[0][1] = -INF;
        for (int i = 1; i <= n; i++)
        {
            f[i][0] = max(f[i - 1][0], f[i - 1][1]);
            f[i][1] = f[i - 1][0] + a[i];
        }
        cout << max(f[n][0], f[n][1]) << endl;
    }
    return 0;
}
```

```python
N = int(1e5) + 10

f = [[0 for _ in range(2)] for i in range(N)]

t = int(input())

for _ in range(t):
    n = int(input())
    a = [0] + list(map(int, input().split()))

    f[0][0] = 0
    f[0][1] = float("-inf")

    for i in range(1, n + 1):
        f[i][0] = max(f[i - 1][0], f[i - 1][1])
        f[i][1] = f[i - 1][0] + a[i]
```

```
    print(max(f[n][0], f[n][1]))
```

1052.设计密码

```
#include <iostream>
#include <algorithm>
#include <cstring>

using namespace std;

const int N = 55, MOD = 1e9 + 7;
int f[N][N], ne[N], ne_map[N][26];
char s[N];

int main()
{
    int n;
    cin >> n >> (s + 1);
    int m = strlen(s + 1);
    f[0][0] = 1;
    for (int i = 2, j = 0; i <= m; i++)
    {
        while (j && s[i] != s[j + 1])
        {
            j = ne[j];
        }
        if (s[i] == s[j + 1])
        {
            j++;
        }
        ne[i] = j;
    }

    for (int j = 0; j < m; j++)
    {
        for (int k = 0; k < 26; k++)
        {
            int u = j;
            while (u && 'a' + k != s[u + 1])
                u = ne[u];
            if ('a' + k == s[u + 1])
                u++;
            ne_map[j][k] = u;
        }
    }

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            for (int k = 0; k < 26; k++)
            {
                int u = ne_map[j][k];
```

```
                if (u < m)
                    f[i + 1][u] = (f[i + 1][u] +  f[i][j]) % MOD;
            }
        }
    }
    int res = 0;
    for (int i = 0; i < m; i++)
        res = (res + f[n][i]) % MOD;

    cout << res << endl;
}
```

```
N = 55
MOD = int(1e9) + 7
ne = [0] * N
f = [[0 for _ in range(N)] for _ in range(N)]
f[0][0] = 1 # 这里f[i][j]表示，已经生成i长度密码，且已经匹配了j个字符的方案，这里不同的i就是不同的状态


n = int(input())

s = ' ' + input()

m = len(s) - 1

j = 0
for i in range(2, m + 1):
    while j and s[i] != s[j + 1]:
        j = ne[j]
    if s[i] == s[j + 1]:
        j += 1
    ne[i] = j

for i in range(n):
    for j in range(m):
        for k in range(ord("a"), ord("z") + 1):
            u = j
            while u and chr(k) != s[u + 1]:
                u = ne[u]
            if chr(k) == s[u + 1]:
                u += 1
            if u < m:
                f[i + 1][u] = (f[i + 1][u] + f[i][j]) % MOD

res = sum(f[n][:m]) % MOD
print(res)
```

1053.修复DNA

```cpp
#include <iostream>
#include <algorithm>
#include <cstring>
#include <cstdio>

using namespace std;

const int N = 1010;
int tr[N][4], ne[N], dar[N], idx;
int q[N], hh = 0, tt = -1;
char s[N];
int f[N][N];

int get(char s)
{
    if (s == 'A')
        return 0;
    if (s == 'T')
        return 1;
    if (s == 'G')
        return 2;
    if (s == 'C')
        return 3;
}

void insert()
{
    int p = 0;
    for (int i = 0; s[i]; i++)
    {
        int t = get(s[i]);
        if (!tr[p][t])
            tr[p][t] = ++idx;
        p = tr[p][t];
    }
    dar[p] = 1;
}

void build()
{
    for (int i = 0; i < 4; i++)
    {
        if (tr[0][i]) // 注意这里
            q[++tt] = tr[0][i];
    }

    while (hh <= tt)
    {
        int t = q[hh++];
        for (int i = 0; i < 4; i++)
        {
            int p = tr[t][i];
            if (!p)
            {
                tr[t][i] = tr[ne[t]][i];
            }
```

```
            else
            {
                ne[p] = tr[ne[t]][i];
                q[++tt] = p;
                dar[p] |= dar[ne[p]];
            }
        }
    }
}

int main()
{
    int t = 1;
    int n;
    while (scanf("%d", &n), n)
    {
        memset(tr, 0, sizeof(tr));
        memset(ne, 0, sizeof(ne));
        memset(dar, 0, sizeof(dar));
        memset(f, 0x3f, sizeof(f));
        idx = 0, hh = 0, tt = -1;
        f[0][0] = 0;
        for (int i = 0; i < n; i++)
        {
            scanf("%s", s);
            insert();
        }
        build();

        scanf("%s", s + 1);
        int m = strlen(s + 1);
        for (int i = 0; i < m; i++)
        {
            for (int j = 0; j <= idx; j++)
            {
                for (int k = 0; k < 4; k++)
                {
                    int t = get(s[i + 1]) != k;
                    int p = tr[j][k];
                    if (!dar[p])
                        f[i + 1][p] = min(f[i + 1][p], f[i][j] + t);
                }
            }
        }

        int res = 0x3f3f3f3f;
        for (int i = 0; i <= idx; i++)
            res = min(res, f[m][i]);
        if (res == 0x3f3f3f3f)
            printf("Case %d: -1\n",t++);
        else
            printf("Case %d: %d\n", t++, res);
    }
}
```

```
N = 1010

tr = [[0 for _ in range(4)] for _ in range(N)]
ne = [0] * N
dar = [0] * N
q = [0] * N
idx = 0
hh = 0
tt = - 1
f = [[float("inf") for _ in range(N)]for _ in range(N)]


def get(c):
    if c == "A":
        return 0
    if c == "T":
        return 1
    if c == "G":
        return 2
    if c == "C":
        return 3


def insert(s):
    global idx
    p = 0
    for i in range(len(s)):
        t = get(s[i])
        if tr[p][t] == 0:
            idx += 1
            tr[p][t] = idx
        p = tr[p][t]
    dar[p] = 1


def build():
    global tt, hh
    for i in range(4):
        if tr[0][i]:
            tt += 1
            q[tt] = tr[0][i]
    while hh <= tt:
        t = q[hh]
        hh += 1
        for i in range(4):
            p = tr[t][i]
            if p == 0:
                tr[t][i] = tr[ne[t]][i]
            else:
                ne[p] = tr[ne[t]][i]
                tt += 1
                q[tt] = p
                dar[p] |= dar[ne[p]]


T = 1
```

```
while True:

    n = int(input())

    if n == 0:
        break
    tr = [[0 for _ in range(4)] for _ in range(N)]
    ne = [0] * N
    dar = [0] * N
    idx = 0
    hh = 0
    tt = -1
    f = [[float("inf") for _ in range(N)]for _ in range(N)]
    f[0][0] = 0
    for i in range(n):
        s = input()
        insert(s)
    build()
    s = ' ' + input().strip() ##某位有空格注意
    m = len(s) - 1
    for i in range(len(s) - 1):
        for j in range(idx + 1):
            for k in range(4):
                cost = 1 if get(s[i + 1]) != k else 0
                p = tr[j][k]
                if dar[p] == 0:
                    f[i + 1][p] = min(f[i + 1][p], f[i][j] + cost)
    res = float("inf")
    for i in range(idx + 1):
        res = min(res, f[m][i])

    if res == float("inf"):
        print(f"Case {T}: -1")
        T += 1
    else:
        print(f"Case {T}: {res}")
        T += 1
```

1057.股票买卖IV

```
#include <algorithm>
#include <cstring>
#include <iostream>

using namespace std;

const int N = 1e5 + 10, K = 110;

int f[K][2];
int a[N];

int main()
```

```
{
    int n, k;
    cin >> n >> k;
    for (int i = 1; i <= n; i++)
        cin >> a[i];
    memset(f, -0x3f, sizeof(f));
    f[0][0] = 0;
    for (int i = 1; i <= n; i++)
    {
        for (int j = k; j >= 1; j--)
        {
            f[j][0] = max(f[j][0], f[j][1] + a[i]);
            f[j][1] = max(f[j][1], f[j - 1][0] - a[i]);
        }
    }
    int res = 0;
    for (int i = 0; i <= k; i++)
        res = max(res, f[i][0]);
    cout << res << endl;
    return 0;
}
```

```
N = int(1e5) + 10
K = 110

f = [[float("-inf") for _ in range(2)] for _ in range(K)]

n, k = map(int, input().split())
a = [0] + list(map(int, input().split()))


f[0][0] = 0

for i in range(1, n + 1):
    for j in range(k, -1, -1):
        f[j][0] = max(f[j][0], f[j][1] + a[i])#  f[j][0]和 f[j][1]的计算顺序不
能反过来
        f[j][1] = max(f[j][1], f[j - 1][0] - a[i])

res = 0
for i in range(k + 1):
    res = max(res, f[i][0])
print(res)
```

1058.股票买卖V

```
#include <iostream>
#include <algorithm>
#include <cstring>
```

```cpp
using namespace std;

const int N = 1e5 + 10;

int f[N][3], a[N];

int main()
{
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++)
        cin >> a[i];
    memset(f, -0x3f, sizeof(f));
    f[0][0] = 0 ;
    for (int i = 1; i <= n; i++)
    {
        f[i][0] = max(f[i - 1][0], f[i - 1][2]);
        f[i][1] = max(f[i - 1][1], f[i - 1][0] - a[i]);
        f[i][2] = f[i - 1][1] + a[i];
    }
    cout << max(f[n][0], f[n][2]) << endl;
    return 0;
}
```

```python
N = int(1e5) + 10

f = [[float("-inf") for _ in range(3)] for _ in range(N)]

n = int(input())
a = [0] + list(map(int, input().split()))
f[0][0] = 0
for i in range(1, n + 1):
    f[i][0] = max(f[i - 1][0], f[i - 1][2])
    f[i][1] = max(f[i - 1][1], f[i - 1][0] - a[i])
    f[i][2] = f[i - 1][1] + a[i]

print(max(f[n][0], f[n][2]))
```

1064.小国王

```cpp
#include <iostream>
#include <algorithm>
#include <cstring>
#include <cstdio>
#include <vector>

using namespace std;

const int N = 12, M = 1 << 10, K = 110;
```

```
typedef long long LL;

LL f[N][K][M];
int cnt[M];
vector<int> state;
vector<int> he[M];
int n, m;

bool check(int state)
{
    if ((state >> 1) & state)
    {
        return false;
    }
    return true;
}

int count(int state)
{
    int res = 0;
    for (int i = 0; i < n; i++)
        res += (state >> i) & 1;
    return res;
}

int main()
{
    scanf("%d%d", &n, &m);
    for (int i = 0; i < (1 << n); i++)
    {
        if (check(i))
        {

            state.push_back(i);
            cnt[i] = count(i);
        }
    }
    for (int i = 0; i < state.size(); i++)
    {
        for (int j = 0; j < state.size(); j++)
        {
            int a = state[i], b = state[j];
            if ((a & b) == 0 && check(a | b))
                he[i].push_back(j);
        }
    }
    f[0][0][0] = 1;
    for (int i = 1; i <= n + 1; i++)
    {
        for (int j = 0; j < m + 1; j++)
        {
            for (int a = 0; a < state.size(); a++)
            {
                for (int b : he[a])
                {
                    int c = cnt[state[a]];
                    if (j >= c)
```

```
                    {
                        f[i][j][state[a]] += f[i - 1][j - c][state[b]];
                    }
                }
            }
        }
    }
    printf("%lld", f[n + 1][m][0]);
    return 0;
}
```

```
N = 12
K = N * N
M = 1 << 10

f = [[[0 for _ in range(M)] for _ in range(K)]for _ in range(N)]
state = []
he = [[] for _ in range(M)]
cnt = [0] * M
n, k = map(int, input().split())


def check(state):
    if state >> 1 & state:
        return False
    return True


def count(state):
    res = 0
    for i in range(n):
        res += state >> i & 1
    return res


for i in range(1 << n):
    if check(i):
        state.append(i)
        cnt[i] = count(i)

for i in range(len(state)):
    for j in range(len(state)):
        a = state[i]
        b = state[j]
        if a & b == 0 and check(a | b):
            he[i].append(j)

f[0][0][0] = 1
for i in range(n + 2):
    for j in range(k + 1):
        for a in range(len(state)):
            for b in range(len(he[a])):
                if j >= cnt[state[a]]:
                    f[i][j][state[a]] += f[i - 1][j -
```

```
                                                    cnt[state[a]][state[he[a]
[b]]]

print(f[n + 1][k][0])
```

11.背包问题求方案数

```
#include <iostream>
#include <cstring>
#include <algorithm>

using namespace std;

const int N = 1010, MOD = 1e9 + 7;

int f[N], g[N];

int main()
{
    int n, m;
    cin >> n >> m;
    for (int i = 0; i <= m; i++)
        g[i] = 1;
    for (int i = 1; i <= n; i++)
    {
        int v, w;
        cin >> v >> w;
        for (int j = m; j >= 0; j--)
        {
            if (j >= v)
            {
                if (f[j] < f[j - v] + w)
                {
                    f[j] = f[j - v] + w;
                    g[j] = g[j - v];
                }
                else if (f[j] == f[j - v] + w)
                {
                    g[j] = (g[j] + g[j - v]) % MOD;
                }
            }
        }
    }

    cout << g[m] << endl;
    return 0;
}
```

```
N = 1010
MOD = int(1e9) + 7
```

```
f = [[0 for _ in range(N)]for _ in range(N)]
g = [[0 for _ in range(N)]for _ in range(N)]
v = [0] * N
w = [0] * N

n, m = map(int, input().split())

for i in range(0, m + 1):
    g[0][i] = 1

for i in range(1, n + 1):
    vi, wi = map(int, input().split())
    v[i] = vi
    w[i] = wi
    for j in range(m, -1, -1):
        f[i][j] = f[i - 1][j]
        if j >= vi:
            if f[i][j] < f[i - 1][j - v[i]] + w[i]:
                f[i][j] = f[i - 1][j - v[i]] + w[i]
                g[i][j] = g[i - 1][j - v[i]]
            elif f[i][j] == f[i - 1][j - v[i]] + w[i]:
                g[i][j] = (g[i - 1][j - v[i]] + g[i - 1][j]) % MOD
            else:
                g[i][j] = g[i - 1][j]
        else:
            g[i][j] = g[i - 1][j]


print(g[n][m])
```

12.背包问题求具体方案

```
#include <iostream>
#include <algorithm>
#include <cstring>

using namespace std;

const int N = 1010;

int f[N][N], v[N], w[N];

int main()
{
    int n, m;
    cin >> n >> m;
    for (int i = 1; i <= n; i++)
    {
        cin >> v[i] >> w[i];
    }
    for (int i = n; i > 0; i--)
    {
```

```
        for (int j = 1; j <= m; j++)
        {
            f[i][j] = f[i + 1][j];
            if (j >= v[i])
                f[i][j] = max(f[i][j], f[i + 1][j - v[i]] + w[i]);
        }
    }

    int j = m;
    for (int i = 1; i <= n; i++)
    {
        if (j >= v[i] && f[i][j] == f[i + 1][j - v[i]] + w[i])
        {
            cout << i << " ";
            j -= v[i];
        }
    }

    return 0;
}
```

```
N = 1010
V = 1010

f = [[0 for _ in range(V)] for _ in range(N)]
v = [0] * N
w = [0] * N


n, m = map(int,input().split())

for i in range(1,n + 1):
    v[i],w[i] = map(int,input().split())

for i in range(n,0,-1):
    for j in range(0,m + 1):
        f[i][j] = f[i + 1][j]
        if j >= v[i]:
            f[i][j] = max(f[i][j],f[i + 1][j - v[i]] + w[i])

j = m

for i in range(1,n + 1):
    if j >= v[i] and f[i][j] == f[i + 1][j - v[i]] + w[i]:
        print(i,end=" ")
        j -= v[i]
```

187.导弹防御系统

```
// 记录全局最小值的做法
// #include <iostream>
// #include <algorithm>
// #include <cstring>

// using namespace std;

// const int N = 55;

// int f[N], g[N], a[N];

// int ans;

// int n;

// void dfs(int u, int su, int sd)
// {
//     if (su + sd >= ans)
//         return;
//     if (u == n)
//     {
//         ans = min(ans, su + sd);
//         return;
//     }

//     int k = 0;
//     while (k < su && f[k] >= a[u])
//         k++;
//     if (k < su)
//     {
//         int t = f[k];
//         f[k] = a[u];
//         dfs(u + 1, su, sd);
//         f[k] = t;
//     }
//     else
//     {
//         f[k] = a[u];
//         dfs(u + 1, su + 1, sd);
//     }

//     k = 0;
//     while (k < sd && g[k] <= a[u])
//     {
//         k++;
//     }
//     if (k < sd)
//     {
//         int t = g[k];
//         g[k] = a[u];
//         dfs(u + 1, su, sd);
//         g[k] = t;
//     }
//     else
//     {
```

```
//         g[k] = a[u];
//         dfs(u + 1, su, sd + 1);
//     }
// }

// int main()
// {

//     while (cin >> n, n)
//     {
//         memset(f, 0, sizeof(f));
//         memset(g, 0, sizeof(g));
//         for (int i = 0; i < n; i++)
//             cin >> a[i];
//         ans = n;
//         dfs(0, 0, 0);
//         printf("%d\n", ans);
//     }

//     return 0;
// }

// 迭代加深的做法

#include <iostream>
#include <algorithm>
#include <cstring>

using namespace std;

const int N = 55;

int f[N], g[N], a[N];

int n;

bool dfs(int depth, int u, int su, int sd)
{
    if (su + sd > depth)
        return false;
    if (u == n)
    {
        return true;
    }

    int k = 0;
    while (k < su && f[k] >= a[u])
        k++;
    if (k < su)
    {
        int t = f[k];
        f[k] = a[u];
        if (dfs(depth, u + 1, su, sd))
            return true;
        f[k] = t;
    }
    else
```

```
    {
        f[k] = a[u];
        if (dfs(depth, u + 1, su + 1, sd))
            return true;
    }

    k = 0;
    while (k < sd && g[k] <= a[u])
    {
        k++;
    }
    if (k < sd)
    {
        int t = g[k];
        g[k] = a[u];
        if (dfs(depth, u + 1, su, sd))
            return true;
        g[k] = t;
    }
    else
    {
        g[k] = a[u];
        if (dfs(depth, u + 1, su, sd + 1))
            return true;
    }
    return false; //注意别忘了
}

int main()
{

    while (cin >> n, n)
    {
        memset(f, 0, sizeof(f));
        memset(g, 0, sizeof(g));
        for (int i = 0; i < n; i++)
            cin >> a[i];
        int depth = 0;
        while (!dfs(depth, 0, 0, 0))
        {
            depth++;
        }
        printf("%d\n", depth);
    }

    return 0;
}
```

```
N = 55

f = [0] * N
g = [0] * N

a = []
```

```
ans = N


def dfs(u, su, sd):
    global ans
    if su + sd >= ans:
        return None
    if u == n:
        ans = min(ans, su + sd)
        return None

    k = 0
    while k < su and f[k] >= a[u]:
        k += 1
    if k < su:
        t = f[k]
        f[k] = a[u]
        dfs(u + 1, su, sd)
        f[k] = t  # 需要回溯
    else:
        f[k] = a[u]
        dfs(u + 1, su + 1, sd)

    k = 0
    while k < sd and g[k] <= a[u]:
        k += 1
    if k < sd:
        t = g[k]
        g[k] = a[u]
        dfs(u + 1, su, sd)
        g[k] = t  # 需要回溯
    else:
        g[k] = a[u]
        dfs(u + 1, su, sd + 1)


while True:
    n = int(input())
    if n == 0:
        break
    a = list(map(int, input().split()))
    f = [0] * N
    g = [0] * N
    ans = n

    dfs(0, 0, 0)
    print(ans)
```

272.最长公共上升子序列

```
#include <iostream>
#include <algorithm>
#include <cstring>
```

```cpp
using namespace std;

const int N = 3010;

int f[N][N], a[N], b[N];

int main()
{
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++)
        cin >> a[i];
    for (int i = 1; i <= n; i++)
        cin >> b[i];

    for (int i = 1; i <= n; i++)
    {
        int maxv = 1;
        for (int j = 1; j <= n; j++)
        {
            f[i][j] = f[i - 1][j];
            if (a[i] == b[j])
                f[i][j] = max(f[i][j], maxv);
            if (b[j] < a[i])
            {
                maxv = max(maxv, f[i - 1][j] + 1);
            }
        }
    }
    int res = 0;
    for (int i = 1; i <= n; i++)
        res = max(res, f[n][i]);
    printf("%d\n", res);
    return 0;
}
```

```python
N = 3010

f = [[0 for _ in range(N)]for _ in range(N)]

a = []
b = []

n = int(input())

a = [0] + list(map(int, input().split()))
b = [0] + list(map(int, input().split()))

res = 0
for i in range(1, n + 1):
    maxv = 1
    for j in range(1, n + 1):
        f[i][j] = f[i - 1][j]
```

```
        if a[i] == b[j]:
            f[i][j] = max(f[i][j], maxv)
        if b[j] < a[i]:
            maxv = max(maxv, f[i - 1][j] + 1)


print(max(f[n]))
```

275.传纸条

```
#include <iostream>
#include <algorithm>
#include <cstring>

using namespace std;

const int N = 55;
int f[2 * N][N][N];
int w[N][N];

int main()
{
    int m, n;
    cin >> m >> n;
    for (int i = 1; i <= m; i++)
    {
        for (int j = 1; j <= n; j++)
        {
            cin >> w[i][j];
        }
    }
    for (int k = 2; k <= m + n; k++)
    {
        for (int i1 = 1; i1 <= m; i1++)
        {
            for (int i2 = 1; i2 <= m; i2++)
            {
                int j1 = k - i1, j2 = k - i2;
                if (j1 >= 1 && j1 <= n && j2 >= 1 && j2 <= n)
                {
                    int t = w[i1][j1];
                    if (i1 != i2)
                    {
                        t += w[i2][j2];
                    }
                    f[k][i1][i2] = max(f[k][i1][i2], f[k - 1][i1 - 1][i2 -
1] + t);
                    f[k][i1][i2] = max(f[k][i1][i2], f[k - 1][i1 - 1][i2] +
t);
                    f[k][i1][i2] = max(f[k][i1][i2], f[k - 1][i1][i2 - 1] +
t);
                    f[k][i1][i2] = max(f[k][i1][i2], f[k - 1][i1][i2] + t);
```

```
                }
            }
        }
    }
    printf("%d\n", f[n + m][m][m]);
    return 0;
}
```

```
N = 55

f = [[[0 for _ in range(N)]for _ in range(N)]for _ in range(2 * N)]
w = []

m,n = map(int,input().split())

w.append([])
for _ in range(m):
    w.append([0] + list(map(int,input().split())))

for k in range(2, n + m + 1):
    for i1 in range(1,m + 1):
        for i2 in range(1,m + 1):
            j1 = k - i1
            j2 = k - i2
            if j1 >= 1 and j1 <= n and j2 >= 1 and j2 <= n:
                t = w[i1][j1]
                if i1 != i2 or k == n + m or k == 2:
                    t += w[i2][j2]
                f[k][i1][i2] = max(f[k][i1][i2], f[k - 1][i1 - 1][i2 - 1] +
t)
                f[k][i1][i2] = max(f[k][i1][i2], f[k - 1][i1 - 1][i2] + t)
                f[k][i1][i2] = max(f[k][i1][i2], f[k - 1][i1][i2 - 1] + t)
                f[k][i1][i2] = max(f[k][i1][i2], f[k - 1][i1][i2] + t)

print(f[n + m][m][m])
```

278.数字组合

```
#include <iostream>
#include <cstring>
#include <algorithm>

using namespace std;

const int N = 105, M = 10000 + 10;

int f[M];
```

```cpp
int main()
{
    int n, m;
    cin >> n >> m;
    f[0] = 1;
    for (int i = 1; i <= n; i++)
    {
        int w;
        cin >> w;
        for (int j = m; j >= w; j--)
        {
            f[j] += f[j - w];
        }
    }
    printf("%d\n", f[m]);
    return 0;
}
```

```python
# 朴素写法
N = 105
M = 10000 + 10

f = [[0 for _ in range(M)]for _ in range(N)]

n, m = map(int, input().split())

a = [0] + list(map(int, input().split()))

for i in range(0,n + 1): ## 注意这里从0开始，从前0个物体在选择也是一种方案
    f[i][0] = 1

for i in range(1, n+1):
    for j in range(1, m + 1):
        f[i][j] = f[i - 1][j]
        if j >= a[i]:
            f[i][j] += f[i - 1][j - a[i]]

print(f[n][m])

# 优化
# N = 105
# M = 10000 + 10

# f = [0 for _ in range(M)]

# n, m = map(int, input().split())

# a = [0] + list(map(int, input().split()))

# f[0] = 1

# for i in range(1, n+1):
#     for j in range(m, 0,-1):
```

```
#         if j >= a[i]:
#             f[j] += f[j - a[i]]

# print(f[m])
```

292.炮兵阵地

```
#include <iostream>
#include <cstring>
#include <cstdio>
#include <algorithm>
#include <vector>

using namespace std;

const int N = 105, M = 1 << 10;

int f[2][M][M];
int cnt[M];
int g[N];

vector<int> state;

int n, m;

int count(int s)
{
    int res = 0;
    for (int i = 0; i < m; i++)
        if (s >> i & 1)
            res += 1;
    return res;
}

int main()
{
    scanf("%d%d", &n, &m);
    for (int i = 1; i <= n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            char c;
            cin >> c;
            if (c == 'H')
                g[i] += 1 << j;
        }
    }
    for (int i = 0; i < 1 << m; i++)
    {
        if (!((i & (i >> 1)) || (i & (i >> 2))))
        {
            state.push_back(i);
            cnt[i] = count(i);
        }
```

```
    }

    for (int i = 1; i <= n; i++)
    {
        for (int j = 0; j < state.size(); j++)
        {
            for (int k = 0; k < state.size(); k++)
            {
                int a = state[j], b = state[k];
                f[i & 1][j][k] = 0; // 养成好习惯，清零
                if ((g[i] & a) || (g[i - 1] & b))
                    continue;
                for (int z = 0; z < state.size(); z++)
                {
                    int c = state[z];
                    // if (g[i - 2] & c)
                    //     continue;
                    if ((a & b) || (a & c) || (b & c))
                        continue;
                    f[i & 1][j][k] = max(f[i & 1][j][k], f[(i - 1) & 1][k]
[z] + cnt[a]);
                }
            }
        }
    }
    int res = 0;
    for (int j = 0; j < state.size(); j++)
        for (int k = 0; k < state.size(); k++)
            res = max(res, f[n & 1][j][k]);
    printf("%d", res);

    return 0;
}
```

```
N = 105
M = 1 << 10

f = [[[0 for _ in range(M)]for _ in range(M)]for _ in range(2)]

state = []
cnt = [0] * M
g = [0] * N


def count(state):
    global m
    res = 0
    for i in range(m):
        if state >> i & 1:
            res += 1
    return res


n, m = map(int, input().split())
```

```python
for i in range(1, n + 1):
    s = input()
    for j in range(m):
        if s[j] == 'H':
            g[i] += (1 << j)

for i in range(1 << m):
    if i & (i >> 1) == 0 and (i & (i >> 2)) == 0:
        state.append(i)
        cnt[i] = count(i)


for i in range(1, n + 1):
    for j in range(len(state)):
        for k in range(len(state)):
            a = state[j]
            b = state[k]
            f[i & 1][j][k] = 0
            if g[i] & a or g[i - 1] & b:
                continue
            for z in range(len(state)):
                c = state[z]
                if (a & b) or (a & c) or (b & c):
                    continue
                f[i & 1][j][k] = max(
                    f[i & 1][j][k], f[(i - 1) & 1][k][z] + cnt[a])

res = 0
for i in range(len(state)):
    for j in range(len(state)):
        a = state[i]
        b = state[j]

        res = max(res, f[n & 1][i][j])
print(res)
```

327.玉米田

```cpp
#include <iostream>
#include <cstring>
#include <algorithm>
#include <cstdio>
#include <vector>

using namespace std;

const int N = 14, M = 1 << 10, MOD = 1e8;

int f[N][M], st[N][N];

vector<int> state[N];
vector<int> he[N][M];
```

```
int m, n;

bool check(int m, int a)
{
    for (int i = 0; i < n; i++)
    {
        if ((st[m][i + 1] >= (a >> i & 1)) && (a >> 1 & a) == 0)
            continue;
        else
            return false;
    }
    return true;
}

int main()
{
    scanf("%d%d", &m, &n);

    for (int i = 1; i <= m; i++)
    {
        for (int j = 1; j <= n; j++)
        {
            scanf("%d", &st[i][j]);
        }
    }
    for (int i = 1; i <= m; i++)
    {
        for (int j = 0; j < 1 << n; j++)
        {
            if (check(i, j))
            {
                state[i].push_back(j);
            }
        }
    }
    for (int i = 2; i <= m; i++)
    {
        for (int j = 0; j < state[i].size(); j++)
        {
            for (int z = 0; z < state[i - 1].size(); z++)
            {
                if ((state[i][j] & state[i - 1][z]) == 0)
                {
                    he[i][state[i][j]].push_back(state[i - 1][z]);
                }
            }
        }
    }
    f[0][0] = 1;
    for (int i = 1; i <= m + 1; i++)
    {
        for (int j : state[i])
        {
            if (i == 1)
            {
                f[i][j] += 1;
            }
```

```
            else
            {
                for (int z : he[i][j])
                {
                    f[i][j] = (f[i - 1][z] + f[i][j]) % MOD;
                }
            }
        }
    }
    int res = 0;
    for (int i = 0; i < 1 << n; i++)
    {

        res = (res + f[m][i]) % MOD;
    }
    printf("%d", res);
    return 0;
}
```

423.采药

```
#include <iostream>
#include <algorithm>
#include <cstring>

using namespace std;

const int T = 1010, M = 105;

typedef pair<int, int> PII;

int f[M][T];
PII a[M];

int main()
{
    int t, m;
    cin >> t >> m;
    for (int i = 1; i <= m; i++)
        cin >> a[i].first >> a[i].second;

    for (int i = 1; i <= m; i++)
    {
        for (int j = 1; j <= t; j++)
        {
            f[i][j] = f[i - 1][j];
            if (j >= a[i].first)
            {
                f[i][j] = max(f[i][j], f[i - 1][j - a[i].first] +
a[i].second);
            }
        }
    }
    printf("%d\n", f[m][t]);
```

```
    return 0;
}
```

```
M = 105
T = 1010

f = [[0 for _ in range(T)]for _ in range(M)]

a = []

t, m = map(int, input().split())

a.append([])
for _ in range(m):
    a.append(list(map(int, input().split())))

for i in range(1, m + 1):
    for j in range(1, t + 1):
        f[i][j] = f[i - 1][j]
        if j >= a[i][0]:
            f[i][j] = max(f[i][j], f[i - 1][j - a[i][0]] + a[i][1])

print(f[m][t])
```

426.开心的金明

```
#include <iostream>
#include <algorithm>
#include <cstring>

using namespace std;

const int N = 30000 + 10;

int f[N], w[N], v[N];

int main()
{
    int n, m;
    cin >> n >> m;
    for (int i = 1; i <= m; i++)
    {
        int vi, pi;
        cin >> vi >> pi;
        v[i] = vi;
        w[i] = vi * pi;
    }

    for (int i = 1; i <= m; i++)
```

```
    {
        for (int j = n; j >= v[i]; j--)
        {
            f[j] = max(f[j], f[j - v[i]] + w[i]);
        }
    }
    cout << f[n] << endl;
    return 0;
}
```

```
N = 30000 + 10
M = 30

f = [0 for _ in range(N)]
w = [0] * N
v = [0] * N
n, m = map(int, input().split())

for i in range(1, m + 1):
    v[i], p = map(int, input().split())
    w[i] = v[i] * p

for i in range(1, m + 1):
    for j in range(n, v[i] - 1, -1):
        f[j] = max(f[j], f[j - v[i]] + w[i])

print(f[n])
```

482.合唱队形

```
#include <iostream>
#include <cstring>
#include <algorithm>

using namespace std;

const int N = 110;

int f[N], g[N], a[N];

int main()
{
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++)
        cin >> a[i];

    for (int i = 1; i <= n; i++)
    {
        f[i] = 1;
```

```
        for (int j = 1; j < i; j++)
        {
            if (a[i] > a[j])
                f[i] = max(f[i], f[j] + 1);
        }
    }
    for (int i = n; i >= 1; i--)
    {
        g[i] = 1;
        for (int j = n; j > i; j--)
        {
            if (a[i] > a[j])
                g[i] = max(g[i], g[j] + 1);
        }
    }
    int res = 0;
    for (int k = 1; k <= n; k++)
        res = max(res, f[k] + g[k] - 1);

    printf("%d\n", n - res);
    return 0;
}
```

```
N = 110

f = [0] * N
g = [0] * N

a = []

n = int(input())

a = [0] + list(map(int, input().split()))

for i in range(1, n + 1):
    f[i] = 1
    for j in range(1, i):
        if a[i] > a[j]:
            f[i] = max(f[i], f[j] + 1)

for i in range(n, 0, - 1):
    g[i] = 1
    for j in range(n, i, -1):
        if a[i] > a[j]:
            g[i] = max(g[i], g[j] + 1)
res = 0
for i in range(1, n + 1):
    res = max(res, f[i] + g[i] - 1)

print(n - res)
```

487.金明的预算方案

```cpp
#include <iostream>
#include <cstring>
#include <algorithm>
#include <vector>

using namespace std;

typedef pair<int, int> PII;

const int N = 60, M = 32010;
PII ms[N];
int f[M];
vector<PII> sv[N];

int main()
{
    int m, n;
    cin >> m >> n;
    for (int i = 1; i <= n; i++)
    {
        int v, p, q;
        cin >> v >> p >> q;
        if (q == 0)
        {
            ms[i] = {v, v * p};
        }
        else
        {
            sv[q].push_back({v, v * p});
        }
    }

    for (int i = 1; i <= n; i++)
    {
        if (ms[i].first == 0)
            continue;
        for (int u = m; u >= 0; u--)
        {
            for (int j = 0; j < 1 << sv[i].size(); j++)
            {
                int v = ms[i].first;
                int w = ms[i].second;
                for (int k = 0; k < sv[i].size(); k++)
                {
                    if (j >> k & 1)
                    {
                        v += sv[i][k].first;
                        w += sv[i][k].second;
                    }
                }
                if (u >= v)
                    f[u] = max(f[u], f[u - v] + w);
            }
        }
    }
    cout << f[m] << endl;
```

```
    return 0;
}
```

```
M = 32010
N = 60
f = [0] * M
ms = [None] * N

sv = [[] for _ in range(N)]


m, n = map(int, input().split())

for i in range(1, n + 1):
    v, p, q = map(int, input().split())
    if q == 0:
        ms[i] = [v, v * p]
    else:
        sv[q].append([v, v * p])

for i in range(1, n + 1):
    if ms[i] is None:
        continue  # 跳过未被赋值的 ms[i]
    for u in range(m, -1, -1):
        for j in range(1 << len(sv[i])):
            v = ms[i][0]
            w = ms[i][1]
            for k in range(len(sv[i])):
                if j >> k & 1:
                    v += sv[i][k][0]
                    w += sv[i][k][1]
            if u >= v:
                f[u] = max(f[u], f[u - v] + w)

print(f[m])
```

524.愤怒的小鸟

```
#include <iostream>
#include <cstring>
#include <cstdio>
#include <algorithm>
#include <cmath>

#define x first
#define y second

using namespace std;
```

```cpp
const int N = 20;
const double eps = 1e-8;

typedef pair<double, double> PDD;
int f[1 << N], path[N][N];
PDD q[N];

int cmp(double x, double y)
{
    if (fabs(x - y) < eps)
        return 0;
    return x < y ? -1 : 1; // 与第二个代码一致
}

int main()
{
    int t;
    scanf("%d", &t);
    while (t--)
    {
        int n, m;
        scanf("%d%d", &n, &m);
        for (int i = 0; i < n; i++)
            scanf("%lf%lf", &q[i].x, &q[i].y);
        memset(path, 0, sizeof(path));
        for (int i = 0; i < n; i++)
        {
            path[i][i] = 1 << i;
            for (int j = 0; j < n; j++)
            {
                double x1 = q[i].x, y1 = q[i].y;
                double x2 = q[j].x, y2 = q[j].y;
                if (!cmp(x1, x2))
                    continue;
                double a = (y1 / x1 - y2 / x2) / (x1 - x2);
                double b = y1 / x1 - a * x1;

                if (cmp(a, 0) >= 0)
                    continue;

                int state = 0;
                for (int k = 0; k < n; k++)
                {
                    double x3 = q[k].x, y3 = q[k].y;
                    if (!cmp(a * x3 * x3 + b * x3, y3))
                        state += 1 << k;
                }
                path[i][j] = state;
            }
        }

        memset(f, 0x3f, sizeof(f));
        f[0] = 0;
        for (int i = 0; i + 1 < 1 << n; i++)
        {
            int x = 0;
            for (int j = 0; j < n; j++)
```

```
            {
                if (!(i >> j & 1))
                {
                    x = j;
                    break;
                }
            }
            for (int j = 0; j < n; j++)
            {
                f[i | path[x][j]] = min(f[i | path[x][j]], f[i] + 1);
            }
        }
        printf("%d\n", f[(1 << n) - 1]);
    }

    return 0;
}
```

```
from math import fabs
N = 20
M = 1 << N
eps = 1e-8

f = [0] * M
q = [[0 for _ in range(2)] for _ in range(N)]
path = [[0 for _ in range(N)] for _ in range(N)]


t = int(input().strip())


def cmp(x, y):
    return fabs(x - y) < eps


for _ in range(t):
    n, m = map(int, input().strip().split())
    for i in range(n):
        x, y = map(float, input().strip().split())
        q[i][0] = x
        q[i][1] = y
    path = [[0 for _ in range(N)] for _ in range(N)]

    for i in range(n):
        path[i][i] = 1 << i
        for j in range(n):
            x1 = q[i][0]
            y1 = q[i][1]
            x2 = q[j][0]
            y2 = q[j][1]
            if cmp(x1, x2):
                continue
            a = (y1 / x1 - y2 / x2) / (x1 - x2)
            b = y1 / x1 - a * x1
```

```
            if a > 0 or cmp(a, 0):
                continue

            state = 0
            for k in range(n):
                x3 = q[k][0]
                y3 = q[k][1]
                if (cmp(a * x3 * x3 + b * x3, y3)):
                    state += 1 << k
            path[i][j] = state

    f = [float("inf")] * M
    f[0] = 0
    for i in range((1 << n) - 1):
        x = 0
        for j in range(n):
            if (i >> j & 1) == 0:
                x = j
                break

        for j in range(n):
            f[i | path[x][j]] = min(f[i | path[x][j]], f[i] + 1)
    print(f[(1 << n) - 1])
```

529.宝藏

```
#include <iostream>
#include <algorithm>
#include <cstring>
#include <cstdio>

using namespace std;

const int N = 15, INF = 0x3f3f3f3f;

int g[N][N], f[1 << N][N], ne[1 << N];

int main()
{
    int n, m;
    cin >> n >> m;
    memset(g, 0x3f, sizeof(g));
    for (int i = 0; i < n; i++)
        g[i][i] = 0;
    for (int i = 0; i < m; i++)
    {
        int a, b, c;
        cin >> a >> b >> c;
        a--; // 转换为0-based
        b--; // 转换为0-based
        g[b][a] = g[a][b] = min(g[a][b], c);
    }
```

```
    for (int i = 0; i < (1 << n); i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (i >> j & 1)
            {
                for (int k = 0; k < n; k++)
                {
                    if (g[j][k] != INF)
                    {
                        ne[i] |= 1 << k;
                    }
                }
            }
        }
    }
    memset(f, 0x3f, sizeof(f));
    for (int i = 0; i < n; i++)
        f[1 << i][0] = 0;
    for (int i = 0; i < 1 << n; i++)
    {
        for (int j = (i - 1) & i; j; j = (j - 1) & i)
        {

            if ((ne[j] & i) == i) // 注意运算符
            {
                int cost = 0;
                int remain = j ^ i;
                for (int k = 0; k < n; k++)
                {
                    if (remain >> k & 1)
                    {
                        int t = INF;
                        for (int u = 0; u < n; u++)
                        {
                            if (j >> u & 1)
                            {
                                t = min(t, g[u][k]);
                            }
                        }
                        cost += t;
                    }
                }
                for (int k = 1; k < n; k++)
                {
                    f[i][k] = min(f[i][k], f[j][k - 1] + cost * k);
                }
            }
        }
    }

    int res = INF;
    for (int i = 0; i < n; i++)
        res = min(res, f[(1 << n) - 1][i]);
    cout << res;
    return 0;
```

```
}
```

532.货币系统

```
#include <iostream>
#include <algorithm>
#include <cstring>

using namespace std;

const int N = 25000 + 10;

int f[N], a[N];

int main()
{
    int t;
    cin >> t;
    while ((t--))
    {
        int n;
        cin >> n;
        memset(f, 0, sizeof(f));
        memset(a, 0, sizeof(f));
        for (int i = 1; i <= n; i++)
            cin >> a[i];
        sort(a + 1, a + n + 1);
        int m = a[n];
        f[0] = 1;
        int res = 0;
        for (int i = 1; i <= n; i++)
        {
            if (f[a[i]] == 0)
                res++;
            for (int j = a[i]; j <= m; j++)
            {
                f[j] = f[j] | f[j - a[i]];
            }
        }
        cout << res << endl;
    }
    return 0;
}
```

```
N = 25000 + 10

f = [0] * N

t = int(input())
```

```python
for _ in range(t):
    f = [0] * N
    n = int(input())

    a = [0] + list(map(int,input().split()))
    a.sort()
    m = a[n]
    f[0] = 1
    res = 0
    for i in range(1,n + 1):
        if f[a[i]] == 0:
            res += 1
        for j in range(a[i],m + 1):
            f[j] |= f[j - a[i]]

    print(res)
```

6.多重背包问题III

```cpp
#include <iostream>
#include <algorithm>
#include <cstring>

using namespace std;

const int V = 20000 + 10;

int f[V], g[V], q[V];
int hh = 0, tt = -1;

int main()
{
    int n, m;
    cin >> n >> m;
    for (int i = 0; i < n; i++)
    {
        int v, w, s;
        cin >> v >> w >> s;
        memcpy(g, f, sizeof(f));
        for (int j = 0; j < v; j++)
        {
            hh = 0, tt = -1;
            for (int k = j; k <= m; k += v)
            {
                if (hh <= tt && q[hh] < k - s * v)
                    hh++;
                while (hh <= tt && g[q[tt]] - (q[tt] - j) / v * w <= g[k] -
(k - j) / v * w)
                {
                    tt--;
                }
                q[++tt] = k;
                f[k] = g[q[hh]] + (k - q[hh]) / v * w;
```

```
            }
        }
    }
    cout << f[m] << endl;
}
```

```
V = 20000 + 10

f = [0] * V
q = [0] * V
hh = 0
tt = -1

n, m = map(int, input().split())

for i in range(1, n + 1):
    v, w, s = map(int, input().split())
    g = f.copy()
    for j in range(v):
        hh = 0
        tt = -1
        for k in range(j, m + 1, v):
            if hh <= tt and q[hh] < k - s * v:
                hh += 1
            while hh <= tt and g[q[tt]] - (q[tt] - j) // v * w <= g[k] - (k
- j) // v * w:
                tt -= 1
            tt += 1
            q[tt] = k
            f[k] = g[q[hh]] + (k - q[hh]) // v * w

print(f[m])
```

7.混合背包问题

```
#include <iostream>
#include <algorithm>
#include <cstring>

using namespace std;

const int N = 1000 + 10;

int f[N];

int main()
{
    int n, m;
    cin >> n >> m;
    while (n--)
```

```
    {
        int v, w, s;
        cin >> v >> w >> s;
        if (!s)
        {
            for (int j = v; j <= m; j++)
            {
                f[j] = max(f[j], f[j - v] + w);
            }
        }
        else
        {
            if (s == -1)
                s = 1;
            for (int k = 1; k <= s; k *= 2)
            {
                for (int j = m; j >= k * v; j--)
                    f[j] = max(f[j], f[j - k * v] + k * w);
                s -= k;
            }
            if (s)
            {
                for (int j = m; j >= s * v; j--)
                    f[j] = max(f[j], f[j - s * v] + s * w);
            }
        }
    }
    cout << f[m] << endl;
    return 0;
}
```

```
V = 1000 + 10
f = [0] * V


n, m = map(int, input().split())

for i in range(n):
    v, w, s = map(int, input().split())
    if s == 0:
        for j in range(v, m + 1):
            f[j] = max(f[j], f[j - v] + w)
    else:
        if s == -1:
            s = 1

        k = 1
        while s >= k:
            for j in range(m, k * v - 1, -1):
                f[j] = max(f[j], f[j - k * v] + k * w)
            s -= k
            k *= 2
        if s:
            for j in range(m, s * v - 1, -1):
                f[j] = max(f[j], f[j - s * v] + s * w)
```

```
print(f[m])
```

734.能量石

```
#include <algorithm>
#include <cstring>
#include <iostream>
#include <cstdio>

using namespace std;

const int N = 105, M = 10000 + 5;

int f[M];

class stone
{
public:
    int s, e, l;

public:
    bool operator<(const stone &other) const
    {
        return this->s * other.l < this->l * other.s;
    }
};

int main()
{
    int t;
    cin >> t;
    for (int k = 1; k <= t; k++)
    {
        memset(f, -0x3f, sizeof(f));
        f[0] = 0;
        stone a[N];
        int n;
        cin >> n;
        int m = 0;
        for (int j = 0; j < n; j++)
        {
            cin >> a[j].s >> a[j].e >> a[j].l;
            m += a[j].s;
        }
        sort(a, a + n);
        for (int i = 0; i < n; i++)
        {
            int s = a[i].s, e = a[i].e, l = a[i].l;
            for (int j = m; j >= s; j--)
            {
                f[j] = max(f[j], f[j - s] + e - (j - s) * l);
            }
```

```
        }
        int res = 0;
        for (int j = 0; j <= m; j++)
            res = max(res, f[j]);
        printf("Case #%d: %d\n", k, res);
    }
    return 0;
}
```

```
N = 10000 + 5

f = [float("-inf")] * N
f[0] = 0


class stone:
    def __init__(self, s, e, l):
        self.s = s
        self.e = e
        self.l = l

    def __lt__(self, stone):
        return self.s * stone.l < stone.s * self.l


t = int(input())

for k in range(1, t + 1):
    f = [float("-inf")] * N
    f[0] = 0
    n = int(input())
    a = []
    m = 0
    for _ in range(n):
        s, e, l = map(int, input().split())
        a.append(stone(s, e, l))
        m += s
    a.sort()

    for i in range(n):
        s, e, l = a[i].s, a[i].e, a[i].l
        for j in range(m, s - 1, -1):
            f[j] = max(f[j], f[j - s] + e - (j - s) * l)

    res = max(f[:m + 1])

    print(f"Case #{k}: {res}")
```

8.二维费用的背包问题

```cpp
#include <iostream>
#include <algorithm>
#include <cstring>

using namespace std;

const int V = 110, M = 110;
int f[V][M];

int main()
{
    int n, v, m;
    cin >> n >> v >> m;
    while (n--)
    {

        int vi, mi, wi;
        cin >> vi >> mi >> wi;
        for (int j = v; j >= vi; j--)
        {
            for (int k = m; k >= mi; k--)
            {
                f[j][k] = max(f[j][k], f[j - vi][k - mi] + wi);
            }
        }
    }
    cout << f[v][m] << endl;
    return 0;
}
```

```python
V = 110
M = 110

f = [[0 for _ in range(M)] for _ in range(V)]

n, v, m = map(int, input().split())

for i in range(n):
    vi, mi, wi = map(int, input().split())
    for j in range(v, vi - 1, -1):
        for k in range(m, mi - 1, -1):
            f[j][k] = max(f[j][k], f[j - vi][k - mi] + wi)

print(f[v][m])
```

第二章搜索

1097.池塘计数

```cpp
#include <iostream>
#include <algorithm>
#include <cstdio>
#include <cstring>

using namespace std;

const int N = 1010;

char g[N][N];
int st[N][N];

int d[][2] = {{-1, -1}, {-1, 0}, {-1, 1}, {0, -1}, {0, 1}, {1, -1}, {1, 0},
{1, 1}};

int n, m;

void dfs(int x, int y)
{
    if (st[x][y])
        return;
    st[x][y] = 1;
    for (int i = 0; i < 8; i++)
    {
        int new_x = d[i][0] + x;
        int new_y = d[i][1] + y;

        if (new_x >= 0 && new_x < n && new_y >= 0 && new_y < m && g[new_x]
[new_y] == 'W')
        {

            dfs(new_x, new_y);
        }
    }
}

int main()
{
    int n, m;
    cin >> n >> m;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            cin >> g[i][j];
        }
    }

    int res = 0;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            if (g[i][j] == 'W' && !st[i][j])
            {
```

```
                res++;
                dfs(i, j);
            }
        }
    }
    cout << res;
    return 0;
}
```

```python
from queue import Queue

q = Queue()

N = 1010

g = []

st = [[0 for _ in range(N)] for _ in range(N)]

d = [[-1, -1], [-1, 0], [-1, 1], [0, -1], [0, 1], [1, -1], [1, 0], [1, 1]]


def bfs(x, y):
    q.put((x, y))
    st[x][y] = 1
    while q.empty() != True:
        t = q.get()
        for i in range(len(d)):
            new_x = d[i][0] + t[0]
            new_y = d[i][1] + t[1]
            if (new_x >= 0 and new_x < n and new_y >= 0 and new_y < m):
                if g[new_x][new_y] == 'W' and st[new_x][new_y] == 0:
                    q.put((new_x, new_y))
                    st[new_x][new_y] = 1


n, m = map(int, input().strip().split())

for i in range(n):
    g.append(input().strip())

cnt = 0
for i in range(n):
    for j in range(m):
        if g[i][j] == 'W' and st[i][j] == 0:
            bfs(i, j)
            cnt += 1

print(cnt)
```

1098.城堡问题

```cpp
#include <algorithm>
#include <cstring>
#include <cstdio>
#include <iostream>
#include <queue>

using namespace std;

const int N = 55;

typedef pair<int, int> PII;

int g[N][N], st[N][N];

int d[][2] = {{0, -1}, {-1, 0}, {0, 1}, {1, 0}};

int m, n;

int bfs(int x, int y)
{
    queue<PII> q; // 创建一个存储整型元素的队列
    q.push({x, y});
    st[x][y] = 1;
    int res = 1;
    while (!q.empty())
    {
        PII t = q.front();
        q.pop();
        int cur_x = t.first, cur_y = t.second;
        for (int i = 0; i < 4; i++)
        {
            if (g[cur_x][cur_y] >> i & 1)
                continue;
            int new_x = cur_x + d[i][0];
            int new_y = cur_y + d[i][1];
            if (new_x >= 0 && new_x < m && new_y >= 0 && new_y < n &&
!st[new_x][new_y])
            {

                q.push({new_x, new_y});
                res++;
                st[new_x][new_y] = 1;
            }
        }
    }
    return res;
}

int main()
{
    cin >> m >> n;
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            cin >> g[i][j];
```

```
        }
    }
    int cnt = 0;
    int max_res = 0;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (!st[i][j])
            {
                cnt += 1;
                max_res = max(max_res, bfs(i, j));
            }
        }
    }
    cout << cnt << endl;
    cout << max_res << endl;
    return 0;
}
```

```
from queue import Queue

N = 55

st = [[0 for _ in range(N)] for _ in range(N)]

d = [[0, -1], [-1, 0], [0, 1], [1, 0]]

g = []


def bfs(x, y):
    q = Queue()
    q.put((x, y))
    st[x][y] = 1
    res = 1
    while q.empty() != True:
        cur_x, cur_y = q.get()

        for i in range(4):
            if g[cur_x][cur_y] >> i & 1:
                continue
            new_x = cur_x + d[i][0]
            new_y = cur_y + d[i][1]
            if new_x >= 0 and new_x < m and new_y >= 0 and new_y < n and
st[new_x][new_y] == 0:
                q.put((new_x, new_y))
                st[new_x][new_y] = 1
                res += 1

    return res


m, n = map(int, input().strip().split())
```

```python
for _ in range(m):
    g.append(list(map(int, input().strip().split())))

cnt = 0
res = 0
for i in range(m):
    for j in range(n):
        if st[i][j] == 0:
            cnt += 1
            res = max(res, bfs(i, j))

print(cnt)
print(res)
```

第四章高级数据结构

1282.搜索关键词

```cpp
#include <iostream>
#include <cstring>
#include <algorithm>
#include <cstdio>

using namespace std;

const int N = 1e4 + 10, M = 1e6 + 10, S = 55;

char s[M];
int tr[N * S][26], q[N * S], idx, ne[N * S], cnt[N * S], hh = 0, tt = -1;

void insert()
{
    int p = 0;
    for (int i = 0; s[i]; i++)
    {
        int u = s[i] - 'a';
        if (!tr[p][u])
            tr[p][u] = ++idx;
        p = tr[p][u];
    }
    cnt[p] += 1;
}

void build()
{
    for (int i = 0; i < 26; i++)
    {
        if (tr[0][i])
            q[++tt] = tr[0][i];
    }
```

```
    while (hh <= tt)
    {
        int t = q[hh++];

        for (int i = 0; i < 26; i++)
        {
            int c = tr[t][i];
            if (!c)
            {
                tr[t][i] = tr[ne[t]][i];
            }
            else
            {
                ne[c] = tr[ne[t]][i];
                q[++tt] = c;
            }
        }
    }
}

int main()
{
    int t;
    scanf("%d", &t);
    while (t--)
    {
        memset(tr, 0, sizeof tr);
        memset(cnt, 0, sizeof cnt);
        memset(ne, 0, sizeof ne);
        idx = 0;
        hh = 0, tt = -1;
        int n;
        scanf("%d", &n);
        for (int i = 0; i < n; i++)
        {
            scanf("%s", s);
            insert();
        }
        build();

        scanf("%s", s);
        int res = 0;
        for (int i = 0, j = 0; s[i]; i++)
        {
            int u = s[i] - 'a';

            j = tr[j][u];
            int p = j;
            while (p && cnt[p] != -1)
            {
                res += cnt[p];
                cnt[p] = -1;
                p = ne[p];
            }
        }
        printf("%d", res);
```

```
    }
}
```

```
N = 10000 + 10
M = 55

tr = [[0 for _ in range(26)] for _ in range(N * M)]
cnt = [0 for _ in range(N * M)]
q = [0 for _ in range(N * M)]
ne = [0 for _ in range(N * M)]
hh = 0
tt = - 1
idx = 0

t = int(input())


def insert(s):
    global idx
    p = 0
    for i in range(len(s)):
        u = ord(s[i]) - ord('a')
        if tr[p][u] == 0:
            idx += 1
            tr[p][u] = idx
        p = tr[p][u]
    cnt[p] += 1


def build():
    global hh, tt
    for i in range(26):
        if tr[0][i]:
            tt += 1
            q[tt] = tr[0][i]

    while hh <= tt:

        t = q[hh]
        hh += 1
        for i in range(26):
            j = ne[t]
            c = tr[t][i]
            if c == 0:
                continue
            while j and tr[j][i] == 0:
                j = ne[j]
            if tr[j][i]:
                j = tr[j][i]
            ne[c] = j
            tt += 1
            q[tt] = c


for _ in range(t):
```

```
tr = [[0 for _ in range(26)] for _ in range(N * M)]
cnt = [0 for _ in range(N * M)]
ne = [0 for _ in range(N * M)]
hh = 0
tt = - 1
idx = 0
n = int(input())
for _ in range(n):
    s = input()
    insert(s)

build()

s = input()
j = 0
res = 0
for i in range(len(s)):
    u = ord(s[i]) - ord('a')

    while j and tr[j][u] == 0:
        j = ne[j]
    if tr[j][u]:
        j = tr[j][u]

    p = j
    while p:
        res += cnt[p]
        cnt[p] = 0
        p = ne[p]

print(res)
```