Assignment #C: 202505114 Mock Exam

Updated 1518 GMT+8 May 14, 2025

2025 spring, Complied by 同学的姓名、院系

> **说明：**
>
> 1. **月考**：AC4（请改为同学的通过数）。考试题目都在“题库（包括计概、数算题目）”里面，按照数字题号能找到，可以重新提交。作业中提交自己最满意版本的代码和截图。
>
> 2. **解题与记录：**
>
>   对于每一个题目，请提供其解题思路（可选），并附上使用Python或C++编写的源代码（确保已在OpenJudge， Codeforces，LeetCode等平台上获得Accepted）。请将这些信息连同显示“Accepted”的截图一起填写到下方的作业模板中。（推荐使用Typora https://typoraio.cn 进行编辑，当然你也可以选择Word。）无论题目是否已通过，请标明每个题目大致花费的时间。
>
> 3. **提交安排：** 提交时，请首先上传PDF格式的文件，并将.md或.doc格式的文件作为附件上传至右侧的“作业评论”区。确保你的Canvas账户有一个清晰可见的头像，提交的文件为PDF格式，并且“作业评论”区包含上传的.md或.doc附件。
>
> 4. **延迟提交：** 如果你预计无法在截止日期前提交作业，请提前告知具体原因。这有助于我们了解情况并可能为你提供适当的延期或其他帮助。
>
> 请按照上述指导认真准备和提交作业，以保证顺利完成课程要求。

1. 题目

E06364: 牛的选举

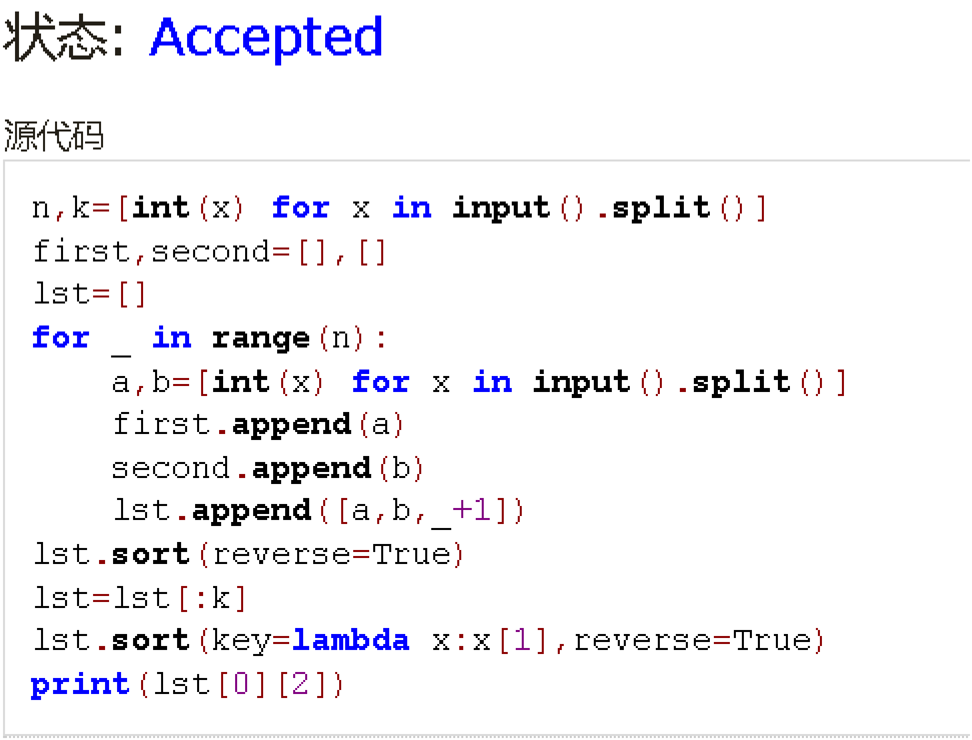
http://cs101.openjudge.cn/practice/06364/

思路：

代码：

```python
n,k=[int(x) for x in input().split()]
first,second=[],[]
lst=[]
for _ in range(n):
    a,b=[int(x) for x in input().split()]
    first.append(a)
    second.append(b)
    lst.append([a,b,_+1])
lst.sort(reverse=True)
lst=lst[:k]
lst.sort(key=lambda x:x[1],reverse=True)
print(lst[0][2])
```

代码运行截图（至少包含有"Accepted"）



状态: Accepted

源代码

```
n,k=[int(x) for x in input().split()]
first,second=[],[]
lst=[]
for _ in range(n):
    a,b=[int(x) for x in input().split()]
    first.append(a)
    second.append(b)
    lst.append([a,b,_+1])
lst.sort(reverse=True)
lst=lst[:k]
lst.sort(key=lambda x:x[1],reverse=True)
print(lst[0][2])
```

M04077: 出栈序列统计

http://cs101.openjudge.cn/practice/04077/

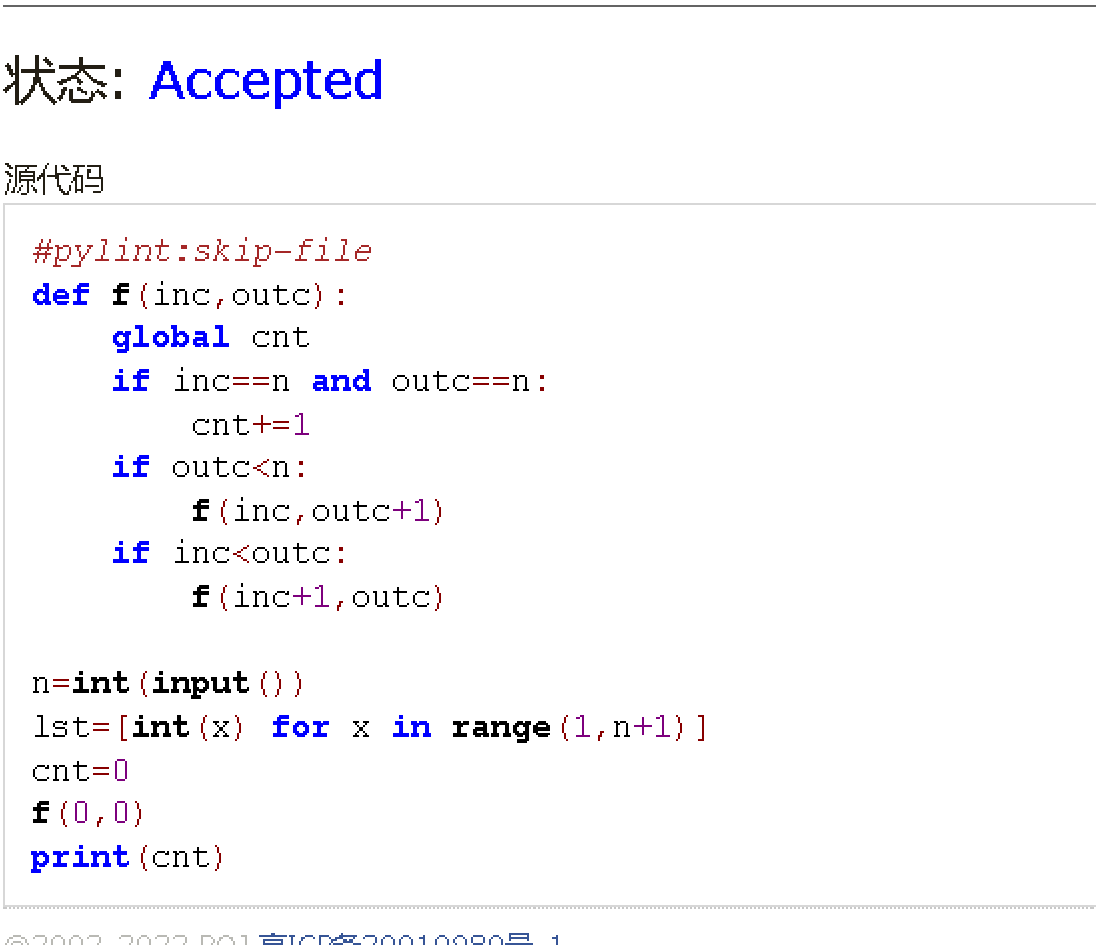
思路：

代码：

```
#pylint:skip-file
def f(inc,outc):
    global cnt
    if inc==n and outc==n:
        cnt+=1
    if outc<n:
        f(inc,outc+1)
    if inc<outc:
        f(inc+1,outc)

n=int(input())
lst=[int(x) for x in range(1,n+1)]
cnt=0
f(0,0)
print(cnt)
```

代码运行截图（至少包含有"Accepted"）



状态: Accepted

源代码

```python
#pylint:skip-file
def f(inc,outc):
    global cnt
    if inc==n and outc==n:
        cnt+=1
    if outc<n:
        f(inc,outc+1)
    if inc<outc:
        f(inc+1,outc)

n=int(input())
lst=[int(x) for x in range(1,n+1)]
cnt=0
f(0,0)
print(cnt)
```

©2002-2022 POJ 京ICP备20010980号-1

M05343:用队列对扑克牌排序

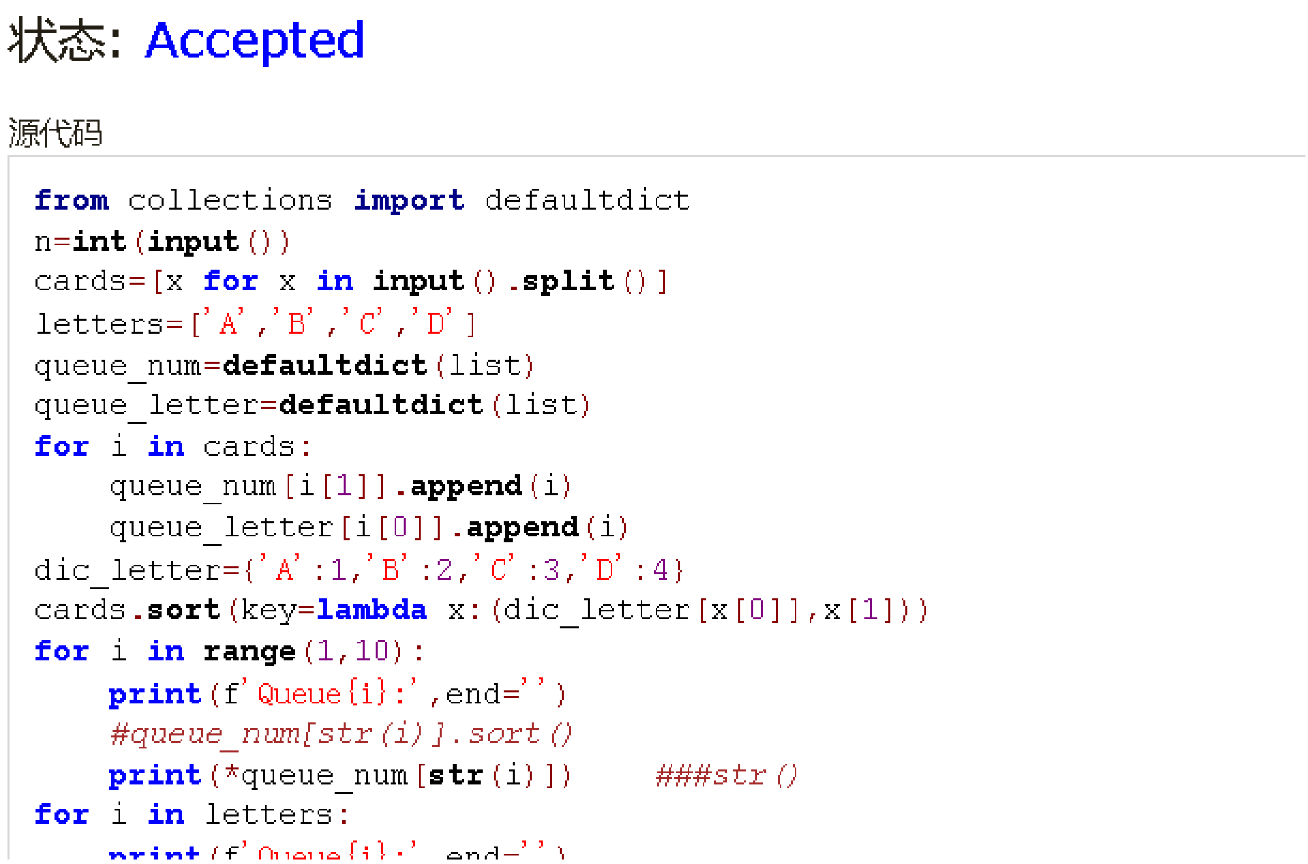
http://cs101.openjudge.cn/practice/05343/

思路：

代码：

```python
from collections import defaultdict
n=int(input())
cards=[x for x in input().split()]
letters=['A','B','C','D']
queue_num=defaultdict(list)
queue_letter=defaultdict(list)
for i in cards:
    queue_num[i[1]].append(i)
    queue_letter[i[0]].append(i)
dic_letter={'A':1,'B':2,'C':3,'D':4}
cards.sort(key=lambda x:(dic_letter[x[0]],x[1]))
for i in range(1,10):
    print(f'Queue{i}:',end='')
    #queue_num[str(i)].sort()
    print(*queue_num[str(i)])     ###str()
for i in letters:
    print(f'Queue{i}:',end='')
    queue_letter[i].sort()
    print(*queue_letter[i])
print(*cards)
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
from collections import defaultdict
n=int(input())
cards=[x for x in input().split()]
letters=['A','B','C','D']
queue_num=defaultdict(list)
queue_letter=defaultdict(list)
for i in cards:
    queue_num[i[1]].append(i)
    queue_letter[i[0]].append(i)
dic_letter={'A':1,'B':2,'C':3,'D':4}
cards.sort(key=lambda x:(dic_letter[x[0]],x[1]))
for i in range(1,10):
    print(f'Queue{i}:',end='')
    #queue_num[str(i)].sort()
    print(*queue_num[str(i)])     ###str()
for i in letters:
    print(f'Queue{i}:',end='')
```

M04084: 拓扑排序

http://cs101.openjudge.cn/practice/04084/

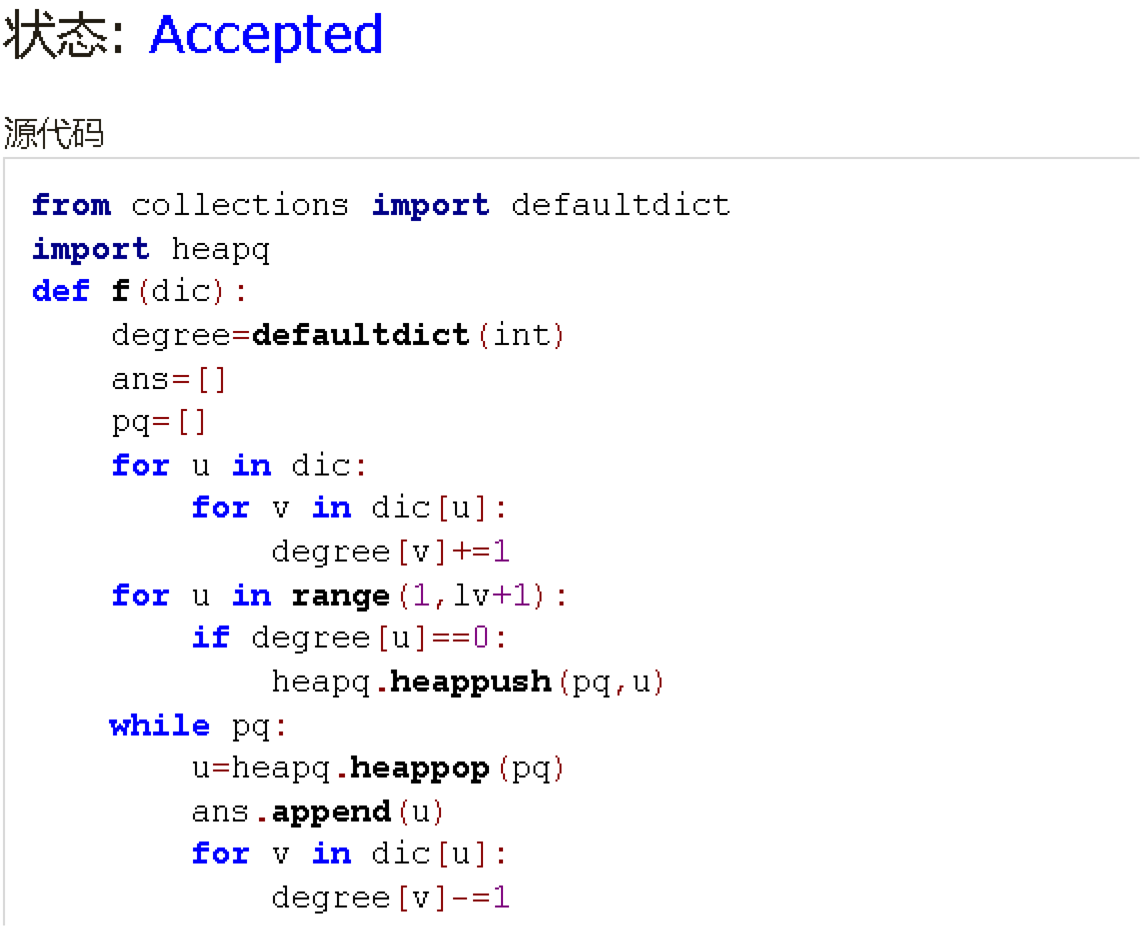
思路：

代码：

```
from collections import defaultdict
import heapq
def f(dic):
    degree=defaultdict(int)
    ans=[]
    pq=[]
    for u in dic:
        for v in dic[u]:
            degree[v]+=1
    for u in range(1,lv+1):
        if degree[u]==0:
            heapq.heappush(pq,u)
    while pq:
        u=heapq.heappop(pq)
        ans.append(u)
        for v in dic[u]:
            degree[v]-=1
            if degree[v]==0:
                heapq.heappush(pq,v)
    return ans

lv,a=[int(x) for x in input().split()]
dic=defaultdict(list)
for _ in range(a):
    m,n=[int(x) for x in input().split()]
    dic[m].append(n)
ans=f(dic)
out=[]
for i in ans:
    out.append('v'+str(i))
print(*out)
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
from collections import defaultdict
import heapq
def f(dic):
    degree=defaultdict(int)
    ans=[]
    pq=[]
    for u in dic:
        for v in dic[u]:
            degree[v]+=1
    for u in range(1,lv+1):
        if degree[u]==0:
            heapq.heappush(pq,u)
    while pq:
        u=heapq.heappop(pq)
        ans.append(u)
        for v in dic[u]:
            degree[v]-=1
```

M07735:道路

Dijkstra, http://cs101.openjudge.cn/practice/07735/

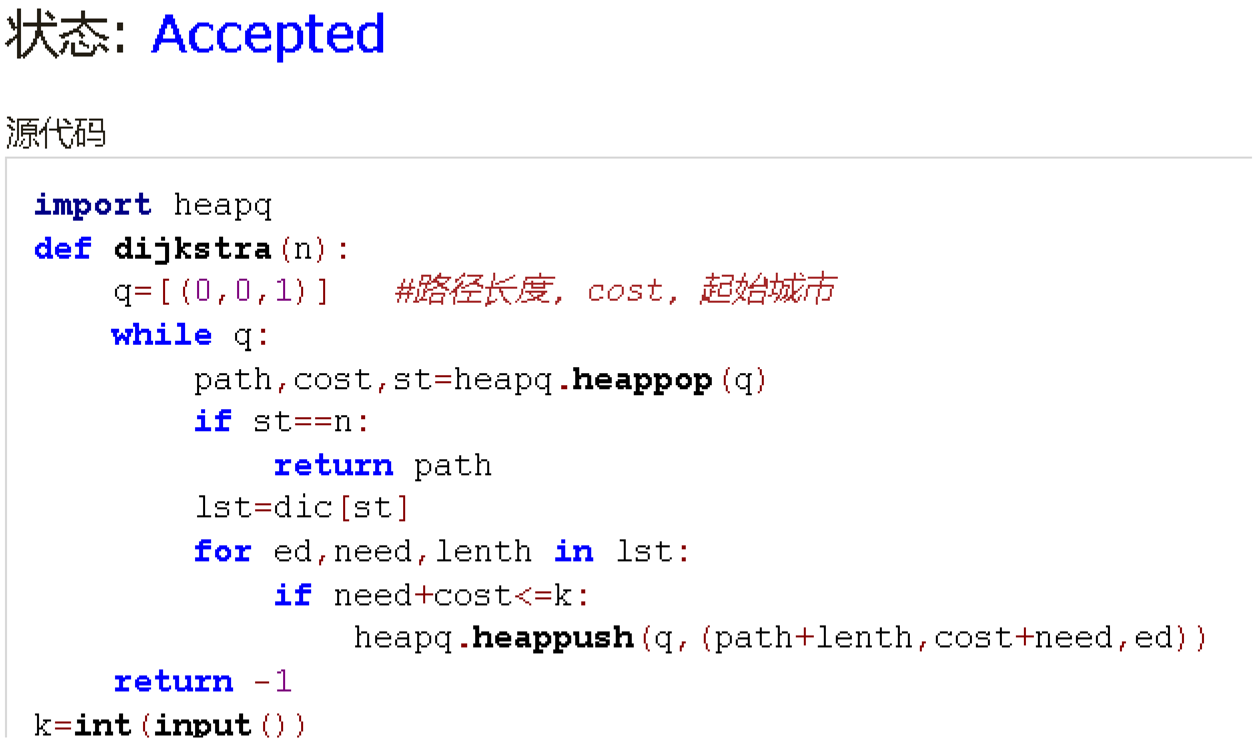
思路：

代码：

```
import heapq
def dijkstra(n):
    q=[(0,0,1)]    #路径长度，cost，起始城市
    while q:
        path,cost,st=heapq.heappop(q)
        if st==n:
            return path
        lst=dic[st]
        for ed,need,lenth in lst:
            if need+cost<=k:
                heapq.heappush(q,(path+lenth,cost+need,ed))
    return -1
k=int(input())
n=int(input())
r=int(input())
dic={int(x):[] for x in range(1,n+1)}
```

```
for _ in range(r):
    s,d,l,t=[int(x) for x in input().split()]
    dic[s].append([d,t,l])   ##ed,cost,len
for i in range(1,n):
    dic[i].sort()
print(dijkstra(n))
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
import heapq
def dijkstra(n):
    q=[(0,0,1)]    #路径长度，cost，起始城市
    while q:
        path,cost,st=heapq.heappop(q)
        if st==n:
            return path
        lst=dic[st]
        for ed,need,lenth in lst:
            if need+cost<=k:
                heapq.heappush(q,(path+lenth,cost+need,ed))
    return -1
k=int(input())
```

T24637:宝藏二叉树

dp, http://cs101.openjudge.cn/practice/24637/

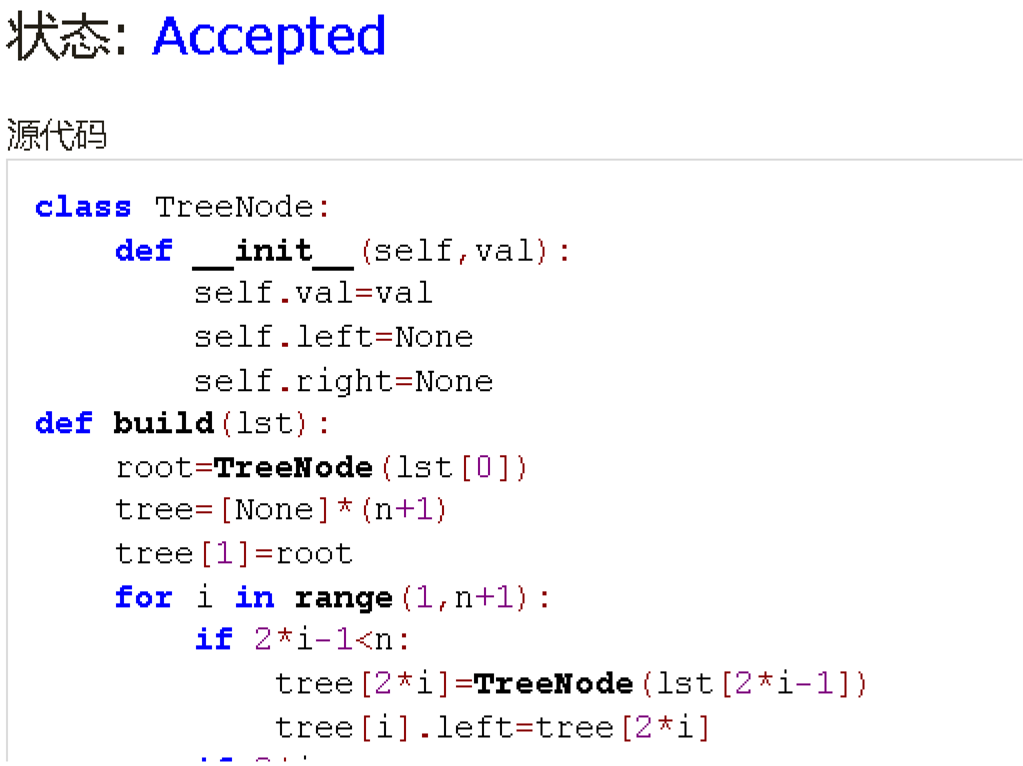
思路：

代码：

```
class TreeNode:
    def __init__(self,val):
        self.val=val
        self.left=None
        self.right=None
def build(lst):
    root=TreeNode(lst[0])
    tree=[None]*(n+1)
    tree[1]=root
    for i in range(1,n+1):
        if 2*i-1<n:
            tree[2*i]=TreeNode(lst[2*i-1])
            tree[i].left=tree[2*i]
        if 2*i<n:
            tree[2*i+1]=TreeNode(lst[2*i])
            tree[i].right=tree[2*i+1]
    return root
def dp(root):
    if not root:
        return 0
    if not root.left or not root.right:
        if not root.left and root.right:
            return max(root.val,root.right.val)
        elif not root.right and root.left:
            return max(root.val,root.left.val)
        else:
            return root.val
    return max(dp(root.left)+dp(root.right),root.val+dp(root.left.left)+dp(root.left.right)+dp(root.right.left)+dp(root.right.right))

n=int(input())
lst=[int(x) for x in input().split()]
root=build(lst)
print(dp(root))
```

代码运行截图（至少包含有"Accepted"）



状态: Accepted

源代码

```
class TreeNode:
    def __init__(self,val):
        self.val=val
        self.left=None
        self.right=None
def build(lst):
    root=TreeNode(lst[0])
    tree=[None]*(n+1)
    tree[1]=root
    for i in range(1,n+1):
        if 2*i-1<n:
            tree[2*i]=TreeNode(lst[2*i-1])
            tree[i].left=tree[2*i]
```

2. 学习总结和收获

如果发现作业题目相对简单，有否寻找额外的练习题目，如“数算2025spring每日选做”、LeetCode、Codeforces、洛谷等网站上的题目。

这次月考没有去机房，自己在寝室考的，前几题因为概念不清楚浪费了很多时间，本来以为最后两题比较难，考试时就直接放弃了，之后写时发现最后两题比前面的简单多了，哎

怎么就要期末了，我感觉我还没准备好啊啊啊啊