

Cheating paper-EXTRA

1.default dict

```
from collections import defaultdict
# 创建一个defaultdict, 值的默认工厂函数为int, 表示默认值为0
char_count = defaultdict(int)
# 统计字符出现次数
input_string = "hello"
for char in input_string:
    char_count[char] += 1
print(char_count) # 输出 defaultdict(<class 'int'>, {'h': 1, 'e': 1, 'l': 2, 'o': 1})
print(char_count['h'])
#输出: 1

dict.get(key,default=None)
```

判断有向图是否有环: 拓扑排序

原理: 每次选入度为0的点, 将该点放入output, 并删掉该点的出边, 同时更新其他点的入度。

如果最后output的长度为n则说明无环

```
class Node:
    def __init__(self, v):
        self.val = v
        self.to = []

from collections import deque

t = int(input())
for _ in range(t):
    n, m = map(int, input().split())
    node = [Node(i) for i in range(1, n + 1)]
    into = [0 for _ in range(n)] # 记录入度
    for _ in range(m): # 构建图
        x, y = map(int, input().split())
        node[x - 1].to.append(node[y - 1])
        into[y - 1] += 1 #更新入度

    queue = deque([node[i] for i in range(n) if into[i] == 0])
    output = []

    while queue:
        a = queue.popleft()
        output.append(a) #把a增加进输出列表中
        for x in a.to: #对于a的指出去的边
            num = x.val
            into[num - 1] -= 1 #删除a指出去的边, 同时更新有向边终点的入度
            if into[num - 1] == 0: #如果更新后入度为0就入队
                queue.append(x)

    if len(output) == n: #如果output的长度是n说明没有环
```

```
print('No')
else:#否则说明有环
    print('Yes')
```

math 库

```
#开头要写
import math
math.sqrt() #开方
math.ceil()#取浮点数上整数
math.floor()#取浮点数的下整数
math.gcd(a,b)#取两个数的最大公约数
math.pow(2,3) # 8.0 幂运算
math.inf#表示正无穷
math.log(100,10) # 2.0
#math.log(x,base) 以base为底，x的对数
math.comb(5,3) # 组合数，C53
math.factorial(5) # 5! 阶乘
```

```
#动态中位数
import heapq
def main():
    lst=list(map(int,input().split()))
    n=len(lst)
    ans=[]
    bigheap=[]
    smallheap=[]
    heapq.heapify(bigheap)
    heapq.heapify(smallheap)
    for i in range(n):
        if not smallheap or -smallheap[0]>=lst[i]:
            heapq.heappush(smallheap,-lst[i])
        else:
            heapq.heappush(bigheap,lst[i])
        if len(bigheap)>len(smallheap):
            heapq.heappush(smallheap,-heapq.heappop(bigheap))
        if len(smallheap)>len(bigheap)+1:
            heapq.heappush(bigheap,-heapq.heappop(smallheap))
        if i%2==0:
            ans.append(-smallheap[0])
    print(len(ans))
    print(' '.join(map(str,ans)))
t=int(input())
for i in range(t):
    main()
```