1. 题目

E22548: 机智的股民老张

http://cs101.openjudge.cn/practice/22548/

代码：

```
a = map(int, input().split())
min_price = float('inf')
max_profit = 0

for price in a:
    min_price = min(min_price, price)
    max_profit = max(max_profit, price - min_price)

print(max_profit)
```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```
a = map(int, input().split())
min_price = float('inf')
max_profit = 0

for price in a:
    min_price = min(min_price, price)   # 更新最小值
    max_profit = max(max_profit, price - min_price)   # 更新最大利润
```

基本信息

#:	47673558
题目:	22548
提交人:	misty
内存:	7856kB
时间:	47ms
语言:	Python3
提交时间:	2024-12-10 22:51:20

M28701: 炸鸡排

greedy, http://cs101.openjudge.cn/practice/28701/

代码：

```
n,k = map(int,input().split())
t = list(map(int,input().split()))
t.sort()
s = sum(t)while True:
    if t[-1] > s/k:
        s -= t[-1]
        t.pop()
        k -= 1
    else:
        print(f'{s / k:.3f}')
        break
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

源代码

```
n,k = map(int,input().split())
t = list(map(int,input().split()))
t.sort()
s = sum(t)
while True:
    if t[-1] > s/k:
        s -= t[-1]
```

基本信息

#: 47673758
题目: 28701
提交人: misty
内存: 3612kB
时间: 27ms
语言: Python3
提交时间: 2024-12-10 23:00:34

M20744: 土豪购物

dp, http://cs101.openjudge.cn/practice/20744/

代码：

```
a=list(map(int,input().split(",")))
n=len(a)
dp1,dp2=a[0],a[0]
ans=0
for i in range(1,n):
    dp1,dp2=max(a[i],dp1+a[i]),max(dp1,dp2+a[i])
    ans=max(ans,dp2)
print(ans)
```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```
a=list(map(int,input().split(",")))
n=len(a)
dp1,dp2=a[0],a[0]
ans=0
for i in range(1,n):
    dp1,dp2=max(a[i],dp1+a[i]),max(dp1,dp2+a[i])
    ans=max(ans,dp2)
```

基本信息

#: 47674079
题目: 20744
提交人: misty
内存: 9412kB
时间: 64ms
语言: Python3
提交时间: 2024-12-10 23:18:53

T25561: 2022 决战双十一

brute force, dfs, http://cs101.openjudge.cn/practice/25561/

代码：

```
result = float("inf")
n, m = map(int, input().split())
store_prices = [input().split() for _ in range(n)]
you= [input().split() for _ in range(m)]
la=[0]*m
def dfs(i,sum1):
    global result
    if i==n:
        jian=0
        for i2 in range(m):
            store_j=0
            for k in you[i2]:
                a,b=map(int,k.split('-'))
                if la[i2]>=a:
```

```
                    store_j=max(store_j,b)
                jian+=store_j
            result=min(result,sum1-(sum1//300)*50-jian)
            return
        for i1 in store_prices[i]:
            idx,p=map(int,i1.split(':'))
            la[idx-1]+=p
            dfs(i+1,sum1+p)
            la[idx-1]-=p
dfs(0,0)
print(result)
```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```
result = float("inf")
n, m = map(int, input().split())
store_prices = [input().split() for _ in range(n)]
you= [input().split() for _ in range(m)]
la=[0]*m
def dfs(i,sum1):
    global result
    if i==n:
```

基本信息

#:	47674112
题目:	25561
提交人:	misty
内存:	3676kB
时间:	66ms
语言:	Python3
提交时间:	2024-12-10 23:20:41

T20741: 两座孤岛最短距离

dfs, bfs, http://cs101.openjudge.cn/practice/20741/

代码：

```
from collections import deque
def dfs(x, y, grid, n, queue, directions):
    """ Mark the connected component starting from (x, y) as visited using DFS. """
    grid[x][y] = 2  # Mark as visited
    queue.append((x, y))
    for dx, dy in directions:
        nx, ny = x + dx, y + dy
        if 0 <= nx < n and 0 <= ny < n and grid[nx][ny] == 1:
            dfs(nx, ny, grid, n, queue, directions)
def bfs(grid, n, queue, directions):
    """ Perform BFS to find the shortest path to another component. """
    distance = 0
    while queue:
        for _ in range(len(queue)):
            x, y = queue.popleft()
            for dx, dy in directions:
                nx, ny = x + dx, y + dy
                if 0 <= nx < n and 0 <= ny < n:
                    if grid[nx][ny] == 1:
                        return distance
                    elif grid[nx][ny] == 0:
```

```
                                grid[nx][ny] = 2    # Mark as visited
                                queue.append((nx, ny))
            distance += 1
    return distance
def main():
    n = int(input())
    grid = [list(map(int, input())) for _ in range(n)]
    directions = [(1, 0), (-1, 0), (0, 1), (0, -1)]
    queue = deque()

    # Start DFS from the first '1' found and use BFS from there
    for i in range(n):
        for j in range(n):
            if grid[i][j] == 1:
                dfs(i, j, grid, n, queue, directions)
                return bfs(grid, n, queue, directions)
if __name__ == "__main__":
    print(main())
```

代码运行截图 （至少包含有"Accepted"）

状态：Accepted

源代码

```
from collections import deque

def dfs(x, y, grid, n, queue, directions):
    """ Mark the connected component starting from (x, y) as visited us
    grid[x][y] = 2  # Mark as visited
    queue.append((x, y))
    for dx, dy in directions:
        nx, ny = x + dx, y + dy
```

基本信息

#: 47674157
题目: 20741
提交人: misty
内存: 4300kB
时间: 29ms
语言: Python3
提交时间: 2024-12-10 23:23:21

T28776: 国王游戏

greedy, http://cs101.openjudge.cn/practice/28776

代码：

```
n = int(input())
a0,b0 = map(int,input().split())
numbers = []
for _ in range(n):
    a,b = map(int,input().split())
    numbers.append((a,b))
numbers.sort(key = lambda x:(x[0]*x[1]))
result = 0
for i in range(n):
    result = max(result,a0 //numbers[i][1])
    a0 *= numbers[i][0]
print(result)
```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```
n = int(input())
a0,b0 = map(int,input().split())
numbers = []
for _ in range(n):
    a,b = map(int,input().split())
    numbers.append((a,b))
numbers.sort(key = lambda x:(x[0]*x[1]))
```

基本信息

#:	47674507
题目:	28776
提交人:	misty
内存:	3616kB
时间:	22ms
语言:	Python3
提交时间:	2024-12-10 23:42:03

2. 学习总结和收获

1 现在比较会的就 dp 了

2 bfsdfs 一长就感觉难学

3 贪心写起来很公式化，就是不好想排序的条件，比如国王游戏