

## Web 付録 第3章 Python プログラム

### プログラム 差別価格オークションと一様価格オークション

以下のプログラムは、複数財の同質財オークションについて、差別価格オークションと一様価格オークションの結果を求めるものです。最初に参加するプレーヤー数  $N$  と配分される財の数  $M$  を設定しています。

次に、各プレーヤーの財に対する評価値を 2 次元リスト (行列) `value` で設定しています。各行が各プレーヤーに対応し、左から順に 1 つ目の財に対する評価値、2 つ目の財に対する評価値、……となっています。続いて、同様にして各プレーヤーの入札価格を 2 次元リスト (行列) `bid` で設定しています。

それから、落札者を決定するために、すべてのプレーヤーの入札価格を 1 次元のリスト (ベクトル) `b` にまとめてから、その値を降順に並べ替えます。その際、それぞれの入札価格を付けたのがどのプレーヤーなのか、その番号を `b_index` に保存しています。ちなみに、並べ替えは、多くのアルゴリズムの教科書で最初に記されている単純なバブル・ソートを使用しています。

最後に、差別価格オークションおよび一様価格オークションにおける落札者とその支払額を画面表示しています。なお、Python では `for` 文のインデックスの番号やリストの要素番号は 0 から始まるため、プレーヤー番号を画面表示する際にはそのインデックスに 1 を足していることに注意してください。

```
# Discriminative price and uniform-price auctions

# プレーヤーの数
N = 2

# 財の数
M = 2

# 評価値 各行に各プレーヤーの 1 つ目の財, 2 つ目の財, ……に対する評価値
value = [
    [80,40],
    [65,30]
]
for i in range(N):
    print('プレーヤー',i+1,'の評価値: ', end="")
```

```

    for j in range(M):
        print(' ',value[i][j],end="")
    print()

print()
# 入札価格 各行に各プレイヤーの1つ目の財, 2つ目の財, ……に対する入札価格
bid = [
    [80,40],
    [65,30]
]
for i in range(N):
    print('プレイヤー',i+1,'の入札価格: ', end="")
    for j in range(M):
        print(' ',bid[i][j],end="")
    print()

print()
# すべてのプレイヤーの入札価格を1つのベクトルにまとめる
b = [0]*(N*M)
b_index = [0]*(N*M)
for i in range(N):
    for j in range(M):
        b[i*M+j] = bid[i][j]
        b_index[i*M+j] = i

# 入札価格を降順に並べ替える (バブル・ソート)
for i in range(N*M):
    for j in range(N*M-1,i,-1):
        if b[j] > b[j-1]:
            b[j], b[j-1] = b[j-1], b[j]
            b_index[j], b_index[j-1] = b_index[j-1], b_index[j]

# 落札者と支払い額の決定
# 差別価格オークションの場合
print('差別価格オークション:')

```

```

for j in range(M):
    print(j+1,'位の落札者: プレーヤー-',b_index[j]+1,' 支払額 = ',b[j])

print()
# 一様価格オークションの場合
print('一様価格オークション:')
for j in range(M):
    print(j+1,'位の落札者: プレーヤー-',b_index[j]+1,' 支払額 = ',b[M])

```

本文の例 3.4 の状況において、各プレーヤーがそれぞれの評価値を正直に入札した場合について、このプログラムを実行した結果は以下のようになります。

```

プレーヤー 1 の評価値:  80  40
プレーヤー 2 の評価値:  65  30

プレーヤー 1 の入札価格:  80  40
プレーヤー 2 の入札価格:  65  30

差別価格オークション:
1 位の落札者: プレーヤー 1  支払額 =  80
2 位の落札者: プレーヤー 2  支払額 =  65

一様価格オークション:
1 位の落札者: プレーヤー 1  支払額 =  40
2 位の落札者: プレーヤー 2  支払額 =  40

```

ここで、本文でも説明したように、プレーヤー1には2つ目の財に対しては評価値 40 以下の価格を入札するインセンティブがあります。例えば、プレーヤー1が2つ目の財について 31 という価格を付ける場合を考えるために、入札価格を以下のように変更してプログラムを再度実行してみます。

```

bid = [
    [80,31],
    [65,30]
]

```

そうして得られた結果は以下のようになります。差別価格オークションの結果は変わりませんが、一様価格オークションの場合、両プレイヤーの支払額が低下していることがわかります。

プレイヤー 1 の評価値: 80 40

プレイヤー 2 の評価値: 65 30

プレイヤー 1 の入札価格: 80 31

プレイヤー 2 の入札価格: 65 30

差別価格オークション:

1 位の落札者: プレイヤー 1 支払額 = 80

2 位の落札者: プレイヤー 2 支払額 = 65

一様価格オークション:

1 位の落札者: プレイヤー 1 支払額 = 31

2 位の落札者: プレイヤー 2 支払額 = 31