Cálculo de complexidade

```
def reduceScore(acc, item):
    if(item['my'] > item['theirs']):
        return acc + 2
    return acc + 1
```

Quantidade	Tipo	Valor
1	if	1

Total
2

```
def getScore(team):
    results = team['data']

    return reduce(reduceScore, results, 0)
```

Quantidade	Tipo	Valor
0		0

Total
1

```
def getAverage(team):
    results = team['data']

    scored = reduce(lambda x, y: x+y['my'], results, 0)
    received = reduce(lambda x, y: x+y['theirs'], results, 0)

    if(received != 0):
        return float(scored) / float(received)
    return float(scored)
```

Quantidade	Tipo	Valor
1	if	1

Total
2

```
def result(filename):
    file = open(filename, "r")

    numberOfTeams = int(file.readline())

    numberOfGames = (numberOfTeams * (numberOfTeams -1))/2

    teamsData = {}

    # Incializa um array pra cada time guardar os resultados do seu jogo.
    # Usando index+1 porque os times virao numerados em base 0.
    for index in range(numberOfTeams):
        teamsData[index+1] = []

    for _ in range(numberOfGames):
        [teamA, scoreA, teamB, scoreB] = map(int, file.readline().split("
"))
        gameA = {'my': scoreA, 'theirs': scoreB}
        gameB = {'my': scoreB, 'theirs': scoreA}
        teamsData[teamA].append(gameA)
        teamsData[teamB].append(gameB)

    teamsDataList = map(
        lambda index: {'data': teamsData[index], 'index': index},
        list(teamsData)
        )

    teamsDataListWithScoreAndAverage = map(
        lambda team: {
            'average': getAverage(team),
            'score': getScore(team),
            'index': team['index']
            },
        teamsDataList
    )

    teamsSortedByAvg = sorted(
        teamsDataListWithScoreAndAverage,
        key= lambda team: team['average']
    )

    teamSortedByScore = sorted(
        teamsSortedByAvg,
        key= lambda team: team['score']
    )

    indexesOrdered = map(lambda team: team['index'], teamSortedByScore)
    return " ".join(list(reversed(map(str, indexesOrdered))))
```

Quantidade	Tipo	Valor

Quantidade	Tipo	Valor
2	for	1

Total
3

Total do programa:

```
2 + 1 + 2 + 3 = 8
```