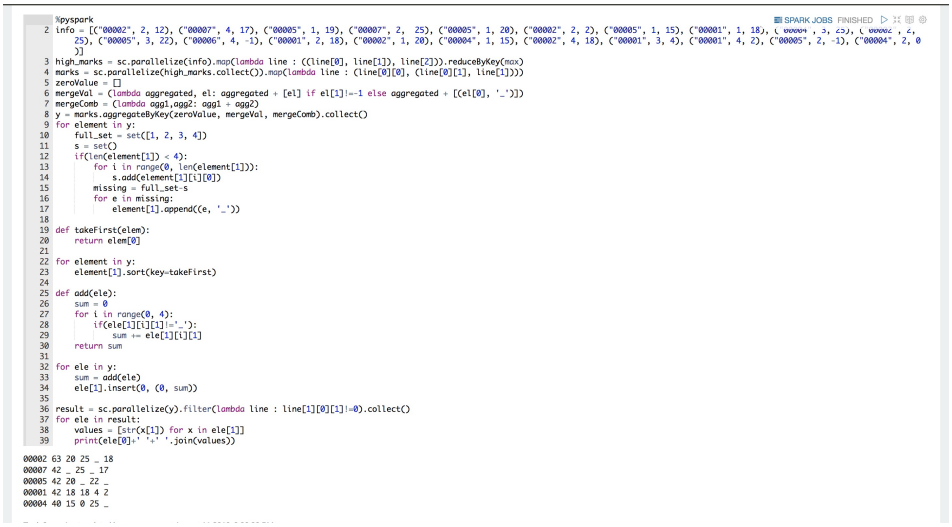
作业2

09_茜

August 2019

1 RDD算子实现PAT Ranking



```
%pyspark
info = [("00002", 2, 12), ("00007", 4, 17), ("00005", 1, 19), ("00007", 2,  25), ("00005", 1, 20), ("00002", 2, 2), ("00005", 1, 15), ("00001", 1, 18), ("00004", 3, 25), ("00005", 3, 22), ("00006", 4, -1), ("00001", 2, 18), ("00002", 1, 20), ("00004", 1, 15), ("00002", 4, 18), ("00001", 3, 4), ("00001", 4, 2), ("00005", 2, -1), ("00004", 2, 0)]
high_marks = sc.parallelize(info).map(lambda line : ((line[0], line[1]), line[2])).reduceByKey(max)
marks = sc.parallelize(high_marks.collect()).map(lambda line : (line[0][0], (line[0][1], line[1])))
zeroValue = []
mergeVal = (lambda aggregated, el: aggregated + [el] if el[1]!=-1 else aggregated + [(el[0], '_')])
mergeComb = (lambda agg1,agg2: agg1 + agg2)
y = marks.aggregateByKey(zeroValue, mergeVal, mergeComb).collect()
for element in y:
    full_set = set([1, 2, 3, 4])
    s = set()
    if(len(element[1]) < 4):
        for i in range(0, len(element[1])):
            s.add(element[1][i][0])
        missing = full_set-s
        for e in missing:
            element[1].append((e, '_'))

def takeFirst(elem):
    return elem[0]

for element in y:
    element[1].sort(key=takeFirst)

def add(ele):
    sum = 0
    for i in range(0, 4):
        if(ele[1][i][1]!='_'):
            sum += ele[1][i][1]
    return sum

for ele in y:
    sum = add(ele)
    ele[1].insert(0, (0, sum))

result = sc.parallelize(y).filter(lambda line : line[1][0][1]!=0).collect()
for ele in result:
    values = [str(x[1]) for x in ele[1]]
    print(ele[0]+' '+' '.join(values))
```

```
00002 63 20 25 _ 18
00007 42 _ 25 _ 17
00005 42 20 _ 22 _
00001 42 18 18 4 2
00004 40 15 0 25 _
```

Figure 1: RDD实现PAT Ranking