此文件用于数据的预处理

In [1]:

```
import os

print('user_artist_data.txt 中数据行数: ')
print(os.popen('cat user_artist_data.txt | wc -l').read())
print('user_artist_data.txt 中数据格式')
print(os.popen('head -5 user_artist_data.txt').read())
```

```
user_artist_data.txt 中数据行数:
 24296858

user_artist_data.txt 中数据格式
1000002 1 55
1000002 1000006 33
1000002 1000007 8
1000002 1000009 144
1000002 1000010 314
```

可以看到 user_artist_data.txt 中共有2400万条数据，其合法的格式应该为 user_id artist_id times ，所以我们利用shell脚本统计一下合理的格式的 数据共有多少。

In [2]:

```
print(os.popen('grep -Ec "^[0-9]+ [0-9]+ [0-9]+$" user_artist_dat
a.txt').read())
```

```
24296858
```

可以看到符合合法格式的数据与数据的总量相同，所以对于 user_aritist_data.txt 这个文件不用进行关于格式的预处理。

In [3]:

```
print('artist_data.txt 中数据的行数')
print(os.popen('cat artist_data.txt | wc -l').read())
print('artist_data.txt 中数据格式')
print(os.popen('head -5 artist_data.txt').read())
# 很奇怪的一点是"似乎在pycharm中可以直接使用python调用shell并且可以使用转义符号"
```

```
artist_data.txt 中数据的行数
 1848579

artist_data.txt 中数据格式
1134999 06Crazy Life
6821360 Pang Nakarin
10113088        Terfel, Bartoli- Mozart: Don
10151459        The Flaming Sidebur
6826647 Bodenstandig 3000
```

类似的来看 `artist_data.txt` ，这个文件中共有180万数据，其合法格式应该为 `artist_id <tab> aritist_name` ，类似的我们使用shell脚本统计一下 合法的数据共有多少。

由于 `<tab>` 在python中的调用使用转义和linux的输入方式均不能得到最终的结果，所以我们直接在终端中使用下面的shell命令来统计相关信息。

```
grep -Ec "^[0-9]+<Control-v><tab>[^<Control-v><tab>^M]+$"
```

其中 `^M` 表示换行符，以上命令的结果为1848063，也就是并不是所有的数据都是符合要求的。

所以我们将数据关于格式的预处理

- 正确格式的数据存放在 `artist_correct_format_data.txt`
- 错误格式的数据存放在 `artist_wrong_format_data.txt`

In [4]:

```
print('artist_alias.txt 中数据的行数')
print(os.popen('cat artist_alias.txt | wc -l').read())
print('artist_data.txt 中数据格式')
print(os.popen('head -5 artist_alias.txt').read())
```

```
artist_alias.txt 中数据的行数
  193027

artist_data.txt 中数据格式
1092764 1000311
1095122 1000557
6708070 1007267
10088054        1042317
1195917 1042317
```

同样对于 artist_alias.txt 文件，其中包含19万余条数据，正确的格式应该为 artist_id <tab> artist_correct_id 。 由于类似的原因，我们也直接使用shell命令进行处理。

```
grep -Ec "^[0-9]+<control-v><tab>[0-9]+$" artist_alia.txt
```

结果为193027，所以存在格式不合法的数据。类似上面的操作。

- 正确格式的数据存放在 artist_correct_format_alias.txt
- 错误格式的数据存放在 artist_wrong_format_alias.txt

In [5]:

```
# 建立从user_id到矩阵下标的映射
user_to_index = dict()
# 建立从artist_id到矩阵下标的映射
artist_to_index = dict()
# user和artist的数量
user_number = 0
artist_number = 0

artist_alias = dict()    # wrong_id, correct_id
artist_data = dict()     # artist_id, artist_name
user_artist = dict()     # (user_index, artist_index), times

with open('./artist_correct_format_alias.txt', 'r') as f:
    for line in f.readlines():
        line = line.replace('\n', '')
        wrong, correct = line.split('\t', 2)
        artist_alias[wrong] = correct

with open('./artist_correct_format_data.txt', 'r') as f:
    for line in f.readlines():
        line = line.replace('\n', '')
        artist_id, artist_name = line.split('\t', 2)
        artist_data[artist_id] = artist_name
        # 如果有重复的artist_id出现，去最后一次出现的值
        if artist_id not in artist_to_index:
            artist_to_index[artist_id] = artist_number
            artist_number = artist_number + 1

with open('./user_artist_data.txt', 'r') as f:
    for line in f.readlines():
        line = line.replace('\n', '')
        user_id, artist_id, times = line.split(' ', 3)
        user_index = -1
        artist_index = -1
        if artist_id in artist_alias:
            artist_id = artist_alias.get(artist_id)

        if artist_id not in artist_to_index:
            continue    # 存在部分artist_id在artist_data.txt中不存在
的现象
        else:
            artist_index = artist_to_index.get(artist_id)

        if user_id in user_to_index:
            user_index = user_to_index.get(user_id)
        else:
            user_to_index[user_id] = user_number
            user_index = user_number
```

```
            user_number = user_number + 1
        # 如果有重复的user_index, artist_index出现，取最后一次有效的值
        user_artist[(user_index, artist_index)] = eval(times)

with open('./out.txt', 'w') as f:
    for item in user_artist.items():
        f.write(str(item[0][0]) + ',' + str(item[0][1]) + ',' + st
r(item[1]) + '\n')

with open('./user_id_to_index.txt', 'w') as f:
    for item in user_to_index.items():
        f.write(item[0] + ',' + str(item[1]) + '\n')

with open('./artist_id_to_index.txt', 'w') as f:
    for item in artist_to_index.items():
        f.write(item[0] + ',' + str(item[1]) + '\n')
```