Report

Data cleaning

我们清洗了原数据中不符合规矩的数据，具体操作是：把rar中的文件解压到 `data` 文件夹，并且：

1. 删除了多余的中文
2. 删除了多余的控制字符
3. 删除了多余的制表符和空行

Reading Data

Rank Info

读取数据的代码包含于主程序中，该代码将一分一段表读取到 `map<int, int>` 中。

```
void readfile(map<int, int>&p, string s) {
    ifstream fin(s, ios::in);
    while (!fin.eof()) {
        int a, b, c;
        fin >> a >> b >> c;
        p[a]=c;
    }
    fin.close();
}
```

Major info

读取专业数据的代码包含于主程序中，该代码将 `data/major.mdb` 读入到 `struct major` 中。在 `struct major` 中，各数据项表示：

```
typedef struct major {
    string name, institute;     //专业名称，开设学院
    bool isScience;             //是否是理科专业
    int min[3];                 //最低分数
    int avr[3];                 //平均分数
} major;
```

读取数据的代码如下：

```
vector<major> readmajor() {
    ifstream fin("data1/major.mdb");
    vector<major> p;
    while (!fin.eof()) {
        major tmp;
        string s, t;
        getline(fin, s);
        s += "\t0\t0\t0\t0\t0\t0";                  //防止缺省的数据
        stringstream str(s);
        str >> tmp.name >> tmp.institute;           //读取专业名称与学院
        str >> t;
        tmp.isScience = (t == "理科");              //是否是理科专业
        int a, b, c, d, e, f;
        str >> a >> b >> c >> d >> e >> f;
        tmp.min[0]=a, tmp.min[1]=c, tmp.min[2]=e;   //读取近三年最低分、平均分
        tmp.avr[0]=b, tmp.avr[1]=d, tmp.avr[2]=f;
        p.push_back(tmp);
    }
    fin.close();
    return p;
}
```

排序

对专业从热门到冷门排序，利用 `algorithm` 中的 `std::sort` 实现。

```
sort(major.begin(), major.end(), [](auto &i, auto &j) {
    if (i.isScience != j.isScience)
        return i.isScience;                             //理科排前面

    return accumulate(i.avr, i.avr+3, 0)/count_if(
            i.avr, i.avr+3, [](auto i){return i;}
        ) >  accumulate(j.avr, j.avr+3, 0)/count_if(
            j.avr, j.avr+3, [](auto i){return i;}
        );                                              //按照近三年平均分排序
});
// TODO: 改进措施：按照近三年的排名信息排序
```

保存信息

将近三年的专业从热门到冷门排序，并补充缺省的排名信息，保存到 `major.csv` 中。由于数据集并未给出文科的一分一段表，故文科专业以后再来探索吧~~

```
ofstream fout("major.csv", ios::out);
for (auto &i: major) {
    fout << i.name << ", " << i.institute << ", ";
    fout << (i.isScience ? "理科, ": "文科, ");
    for (auto j = 0; j < 3; ++j)
        fout << i.min[j] << ", " << i.avr[j] << ", ";
    if (i.isScience)
        for (auto j = 0; j < 3; ++j)
            fout << rank[j][i.min[j]] << ", " << rank[j][i.avr[j]] << ", ";
    fout << endl;
}
fout.close();
```

报考指南

输入分数，按照冲、稳、保三个方面给出报考建议。

1. **冲**：三年中有一年的*最低排名*低于我的排名并且不符合**稳**的条件
2. **稳**：三年的*平均排名*的平均值低于我的排名并且不符合**保**的条件
3. **保**：我的排名高于三年中每一年的*平均排名*。

代码如下：

```
cout << "冲: \n";
for (auto &i: major) {
    auto k = max(max(rank[0][i.min[0]], rank[1][i.min[1]]), rank[2][i.min[2]]);
    auto l = (rank[0][i.avr[0]]+rank[1][i.avr[1]]+rank[2][i.avr[2]])/count_if(i.avr, i.avr+3, [](auto i){return i;});
    if (i.isScience && k >= rank[3][score] && l < rank[3][score]) {
        cout << i.name << " " << i.institute << " ";
        for (auto j = 0; j < 3; ++j)
            cout << i.min[j] << " " << i.avr[j] << " ";
        for (auto j = 0; j < 3; ++j)
            cout << rank[j][i.min[j]] << " " << rank[j][i.avr[j]] << " ";
        cout << endl;
    }
}
cout << "稳: \n";
for (auto &i: major) {
    auto l = (rank[0][i.avr[0]]+rank[1][i.avr[1]]+rank[2][i.avr[2]])/count_if(i.avr, i.avr+3, [](auto i){return i;});
    auto p = MAX;
    for (auto j = 0; j < 3; ++j)
        if (rank[j][i.avr[j]] && rank[j][i.avr[j]] < p)
            p = rank[j][i.avr[j]];
    if (i.isScience && l >= rank[3][score] && p < rank[3][score]) {
        cout << i.name << " " << i.institute << " ";
        for (auto j = 0; j < 3; ++j)
            cout << i.min[j] << " " << i.avr[j] << " ";
        for (auto j = 0; j < 3; ++j)
            cout << rank[j][i.min[j]] << " " << rank[j][i.avr[j]] << " ";
        cout << endl;
    }
}
cout << "保: \n";
for (auto &i: major) {
    int p = MAX;
    for (auto j = 0; j < 3; ++j)
        if (rank[j][i.avr[j]] && rank[j][i.avr[j]] < p)
            p = rank[j][i.avr[j]];
    if (i.isScience && p >= rank[3][score]) {
        cout << i.name << " " << i.institute << " ";
        for (auto j = 0; j < 3; ++j)
            cout << i.min[j] << " " << i.avr[j] << " ";
        for (auto j = 0; j < 3; ++j)
            cout << rank[j][i.min[j]] << " " << rank[j][i.avr[j]] << " ";
        cout << endl;
    }
}
```