### 问题建模

最少费用购物问题与0-1背包问题的逆问题类似，0-1背包问题是给出n种物品和一个背包，使得装入背包中的物品价值最大，背包问题制定了背包的最大容量，求不超过背包最大容量的，且价值最大的最优解。本题则是通过对用户输入想购买的商品及数量，计算出花费最少的购物方案及费用，所谓购物方案，指从哪个网站的哪家店铺购买多少什么商品，主要考虑京东、淘宝、天猫、沃尔玛线上等人们常用的电子商务网站，考虑到小组自身知识欠缺，利用JavaScript 不利于进行数据的跨域爬取操作，所以利用Android Studio进行APP的编写来实现基本操作。

对于最少购物问题的理解，我们可以假设：商店中每种商品都有标价。例如，一朵花的价格是 2 元。一个花瓶的价格是 5 元。为了吸引顾客，商店提供了一组优惠商品价。优惠商品是把一种或多种商品分成一组，并降价销售。例如，3 朵花的价格不是 6 元而是 5 元。2 个花瓶加 1 朵花的优惠价是 10 元。此时需要设计一个算法，计算出某一顾客所购商品应付的最少费用。

**建模条件：**

1、该问题不考虑商品质量问题，均为网上虚拟物品；

2、问题的商品数量无限，无阈值；

3、所有已进行降价销售的组合，不再用作另外的捆绑销售。

**具体思路：**

1. 从网站爬取相关商品数据作为算法的输入
2. 设计算法并用Java代码实现
3. 设计基本的UI界面
4. 整合算法代码到Android Studio中，数据先是自定义输入，成功后再实现动态输入
5. 整合爬虫代码获取的的数据作为输入数据
6. 调试无误后生成APP

**实施流程：**

李华（组长）：负责整合代码、app的实现

林佩怡：负责算法分析与实现

王永丽：讲解代码并录屏，查询各类资料

覃富星、韦茂振：代码实现爬取数据功能

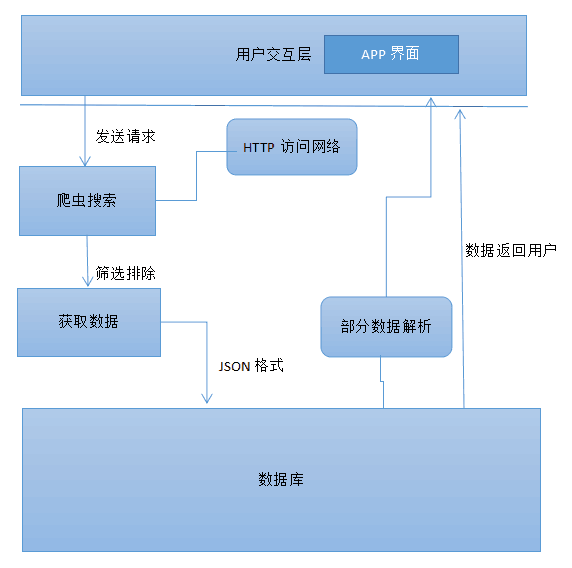
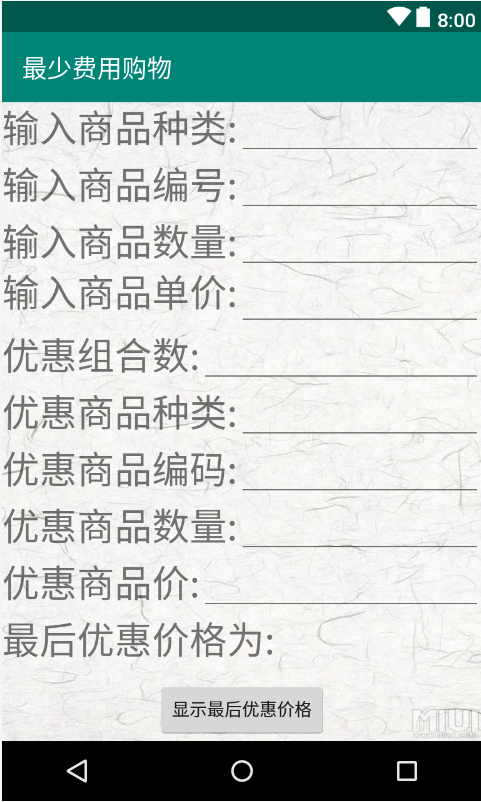
江宏健：完成各类实验文档

编写算法代码

编写爬虫代码，爬取数据，作为算法的输入数据

设计APP界面，以及编写对应逻辑

### 软件框架设计



### 算法设计

对于给定欲购商品的价格和数量，以及优惠商品价，计算所购商品应付的最少费用，

其中相关商品数据如单价、优惠商品价等，通过爬虫功能从各大网站获取

### 算法分析

最少购物费用问题：

数据输入： 由文件 input.txt 提供欲购商品数据。文件的第 1 行中有 1 个整数 B(0≤B≤5)，表示所购商品种类数。接下来的 B 行，每行有 3 个数 C，K 和 P。C 表示商品的编码(每种商品有 唯一编码)，1≤C≤999。K 表示购买该种商品总数，1≤K≤5。P 是该种商品的正常单价(每件商品的价格)，1≤P≤999。请注意，一次最多可购买 5\*5=25 件商品。

由文件 offer.txt 提供优惠商品价数据。文件的第 1 行中有 1 个整数 S(0≤S≤99)，表示 共有 S 种优惠商品组合。接下来的 S 行，每行的第一个数描述优惠商品组合中商品的种类数 j。接着是 j 个数字对(C，K)，其中 C 是商品编码，1≤C≤999。K 表示该种商品在此组合中的数量，1≤K≤5。每行最后一个数字 P(1≤ P≤9999)表示此商品组合的优惠价。

结果输出：将计算出的所购商品应付的最少费用输出到文件output.txt

输入文件示例 输出文件示例

Input.txt offer.txt output.txt

2 2 14

7 3 2 1 7 3 5

8 2 5 2 7 1 8 2 10

### 算法实现

**最少购物问题(代码)：**

import java.util.Scanner;

class Commodity{

int piece;//购买数量

int price;//购买价格

}

public class gwwt{

private static int MAXCODE = 999;//商品编码的最大值

private static int SALECOMB = 99;//优惠商品组合数

private static int KIND = 5; //商品种类

private static int QUANTITY = 5; //购买某种商品数量的最大值

private static int b;//购买商品种类数

private static int s;//当前优惠组合数

private static int[] num = new int[MAXCODE+1];//记录商品编码与商品种类的对应关系

private static int[] product = new int[KIND+1];//记录不同种类商品的购买数量

private static int[][] offer = new int[SALECOMB+1][KIND+1];//offer[i][j]: 商品组合的优惠价(j=0)；某种优惠组合中某种商品需要购买的数量(j>0)

private static Commodity[] purch = new Commodity[KIND+1];//记录不同商品的购买数量和购买价格

private static int[][][][][] cost = new int[QUANTITY+1][QUANTITY+1][QUANTITY+1][QUANTITY+1][QUANTITY+1];//记录本次购买的总花费

public static void main(String[] args){

init();//输入数据

comp(1);//计算优惠

out();//输出最后价格

}

private static void minicost(){

int i,j,k,m,n,p,minm;

minm = 0;

for(i=1; i<=b; i++)

minm += product[i]\*purch[i].price;

for(p=1; p<=s; p++){

i = product[1] - offer[p][1];

j = product[2] - offer[p][2];

k = product[3] - offer[p][3];

m = product[4] - offer[p][4];

n = product[5] - offer[p][5];

if(i>=0 && j>=0 && k>=0 && m>=0 && n>=0 && cost[i][j][k][m][n]+offer[p][0] < minm)

minm = cost[i][j][k][m][n] + offer[p][0];

}

cost[product[1]][product[2]][product[3]][product[4]][product[5]] = minm;

}

private static void init(){

Scanner input = new Scanner(System.in);

int i,j,n,p,t,code;

for(i=0; i<100; i++)

for(j=0; j<6; j++)

offer[i][j] = 0;

for(i=0; i<6; i++){

purch[i] = new Commodity();

purch[i].piece = 0;

purch[i].price = 0;

product[i] = 0;

}

System.out.print("请输入需要购买商品的种类数:");

b = input.nextInt();//输入商品种类数

for(i=1; i<=b; i++){

System.out.print("请输入需要购买商品的编码:");

code = input.nextInt();//商品编码

System.out.print("请输入需要购买商品的数量:");

purch[i].piece = input.nextInt();//购买数量

System.out.print("请输入需要购买商品的单价:");

purch[i].price = input.nextInt();//购买单价

num[code] = i;

}

System.out.print("请输入当前优惠组合数:");

s = input.nextInt();//当前优惠组合数

for(i=1; i<=s; i++){

System.out.print("请输入优惠商品组合中商品种类数:");

t = input.nextInt();//优惠商品组合中商品种类数

for(j=1; j<=t; j++){

System.out.print("请输入优惠商品组合中商品编码:");

n = input.nextInt();//优惠商品编码

System.out.print("请输入优惠商品组合中商品数量:");

p = input.nextInt();//优惠商品数量

offer[i][num[n]] = p;

}

System.out.print("请输入该组合的优惠价格:");

offer[i][0] = input.nextInt();//优惠价格

}

}

private static void comp(int i){

if(i > b){

minicost();

return;

}

for(int j=0; j<=purch[i].piece; j++){

product[i] = j;

comp(i+1);

}

}

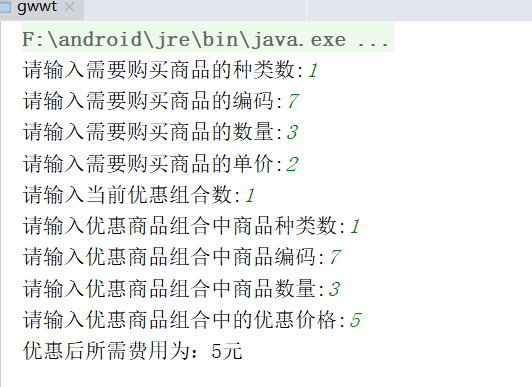
private static void out(){

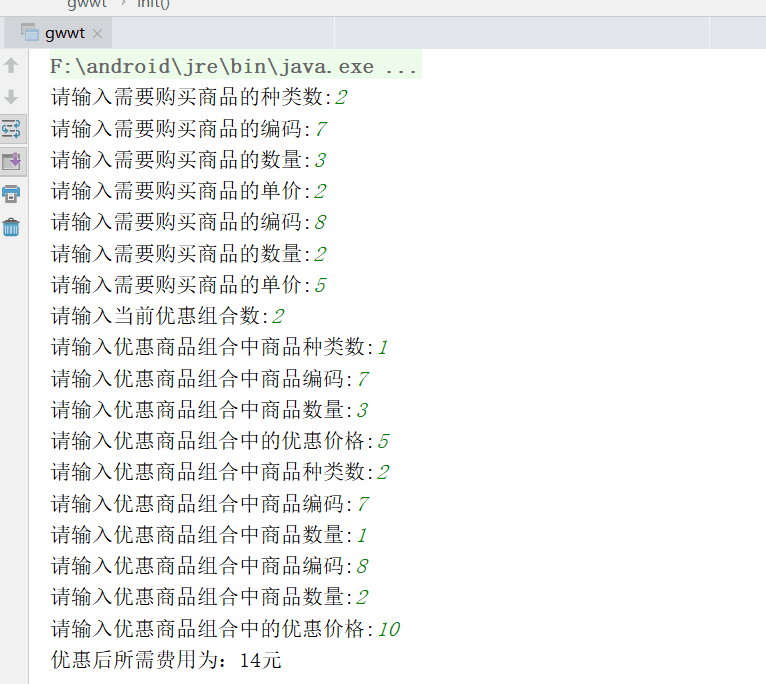
System.out.println("所购买的商品最后优惠价格为:"+cost[product[1]][product[2]][product[3]][product[4]][product[5]]+"元");

}

}

**实例测试：**





**未完成原因：**

一开始两天一直在解决如何用纯js爬取网站数据，因为没接触过js以及不懂得解决跨域问题，很难实现，所以第三天决定换思路，用Android studio做一个APP，所以短时间内未能完整完成实践任务。

**遇到的困难：**

1. 由于没接触过JavaScript短时间内对于JavaScript掌握不到位，无法通过其进行价格数据的准确爬取以及利用。相关代码如下

<!DOCTYPE HTML>

<html>

<body>

<div> 请输入产品型号 <input type="text" id="pname" name="user\_mail" value="54" /></div>

<div> 请输入数量 <input type="text" id="pnum" value="10"> </div>

<div> <button type="button">添加一个商品 <tton> </div>

<div id="mycontainior" <div>

<script>

function addProductItem(){

v str1 ='<div> 请输入商品型号 <input type="text" id="pname" value="54"> '

str1+='<div>请输入数量 <input type="text" id="pnum" value="10">'

document.getElementById(str1).innerHTML=name;

}

function findCheapSolution(items){//商品名称

keyword = items[0].name;

keyword.encode('GB2312');

url = 'https://search.jd.com/Search?keyword='+keyword + '&enc=utf-8&wq='+keyword;

}

var httpRequest = new XMLHttpRequest();//第一步：建立所需的对象

httpRequest.open('GET', 'url', true);//第二步：打开连接 将请求参数写在url中

httpRequest.send();//第三步：发送请求 将请求参数写在URL中

/\*\*

\* 获取数据后的处理程序

\*/

httpRequest.onreadystatechange = function () {

if (httpRequest.readyState == 4 && httpRequest.status == 200) {

var json = httpRequest.responseText;//获取到json字符串，还需解析

console.log(json);

}

};

</script>

</body>

</body>

</html>

1. 在运用Android Studio结合JAVA算法代码的时候，由于这个算法是需要用户动态提供输入数据的，输入数据的量是不确定的，在Android Studio中用EditText这个控件来获取用户输入的数据，每一个量就需要一个EditText控件，所以用这个控件无法实现不定量的动态输入，代码如下

**public class** MainActivity **extends** AppCompatActivity {  
 **private** EditText **editText1**;  
 **private** EditText **editText2**;  
 **private** EditText **editText3**;  
 **private** EditText **editText4**;  
 **private** EditText **editText5**;  
 **private** EditText **editText6**;  
 **private** EditText **editText7**;  
 **private** EditText **editText8**;  
 **private** EditText **editText9**;  
 **private** TextView **jieguo**;  
 **private** Button **button**;  
  
 **private int MAXCODE** = 999;*//商品编码的最大值* **private int SALECOMB** = 99;*//优惠商品组合数* **private int KIND** = 5; *//商品种类* **private int QUANTITY** = 5; *//购买某种商品数量的最大值* **private int b**;*//购买商品种类数* **private int s**;*//当前优惠组合数* **private int**[] **num** = **new int**[**MAXCODE**+1];*//记录商品编码与商品种类的对应关系* **private int**[] **product** = **new int**[**KIND**+1];*//记录不同种类商品的购买数量* **private int**[][] **offer** = **new int**[**SALECOMB**+1][**KIND**+1];*//offer[i][j]: 商品组合的优惠价(j=0)；某种优惠组合中某种商品需要购买的数量(j>0)* **private** Commodity[] **purch** = **new** Commodity[**KIND**+1];*//记录不同商品的购买数量和购买价格* **private int**[][][][][] **cost** = **new int**[**QUANTITY**+1][**QUANTITY**+1][**QUANTITY**+1][**QUANTITY**+1][**QUANTITY**+1];*//记录本次购买的总花费* @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 **editText1**=(EditText)findViewById(R.id.***输入商品种类***);  
 **editText2**=(EditText)findViewById(R.id.***输入商品编号***);  
 **editText3**=(EditText)findViewById(R.id.***输入商品数量***);  
 **editText4**=(EditText)findViewById(R.id.***输入商品单价***);  
 **editText5**=(EditText)findViewById(R.id.***优惠组合数***);  
 **editText6**=(EditText)findViewById(R.id.***优惠商品种类***);  
 **editText7**=(EditText)findViewById(R.id.***优惠商品编码***);  
 **editText8**=(EditText)findViewById(R.id.***优惠商品数量***);  
 **editText9**=(EditText)findViewById(R.id.***优惠商品价***);  
 **jieguo**=(TextView) findViewById(R.id.***结果***);  
 **button**=(Button)findViewById(R.id.***点击***);  
 init();  
 comp(1);  
 out();  
 }  
 **private void** minicost(){  
 **int** i,j,k,m,n,p,minm;  
 minm = 0;  
 **for**(i=1; i<=**b**; i++)  
 minm += **product**[i]\***purch**[i].**price**;  
  
 **for**(p=1; p<=**s**; p++){  
 i = **product**[1] - **offer**[p][1];  
 j = **product**[2] - **offer**[p][2];  
 k = **product**[3] - **offer**[p][3];  
 m = **product**[4] - **offer**[p][4];  
 n = **product**[5] - **offer**[p][5];  
 **if**(i>=0 && j>=0 && k>=0 && m>=0 && n>=0 && **cost**[i][j][k][m][n]+**offer**[p][0] < minm)  
 minm = **cost**[i][j][k][m][n] + **offer**[p][0];  
 }  
 **cost**[**product**[1]][**product**[2]][**product**[3]][**product**[4]][**product**[5]] = minm;  
 }  
 **private void** init(){  
  
 **int** i,j,n,p,t,code;  
 **for**(i=0; i<100; i++)  
 **for**(j=0; j<6; j++)  
 **offer**[i][j] = 0;  
  
 **for**(i=0; i<6; i++){  
 **purch**[i] = **new** Commodity();  
 **purch**[i].**piece** = 0;  
 **purch**[i].**price** = 0;  
 **product**[i] = 0;  
 }  
  
 **b** = Integer.*parseInt*(**editText1**.getText().toString());*//输入商品种类* **for**(i=1; i<=**b**; i++){  
 code = Integer.*parseInt*(**editText2**.getText().toString());*//商品编码* **purch**[i].**piece** = Integer.*parseInt*(**editText3**.getText().toString());*//购买数量* **purch**[i].**price** = Integer.*parseInt*(**editText4**.getText().toString());*//购买单价* **num**[code] = i;  
 }  
  
 **s** = Integer.*parseInt*(**editText5**.getText().toString());*//当前优惠组合数* **for**(i=1; i<=**s**; i++){  
 t = Integer.*parseInt*(**editText6**.getText().toString());*//优惠商品组合中商品种类* **for**(j=1; j<=t; j++){  
 n = Integer.*parseInt*(**editText7**.getText().toString());*//优惠商品编码* p = Integer.*parseInt*(**editText8**.getText().toString());*//优惠商品数量* **offer**[i][**num**[n]] = p;  
 }  
 **offer**[i][0] = Integer.*parseInt*(**editText9**.getText().toString());*//优惠价格* }  
 }  
 **private void** comp(**int** i){  
 **if**(i > **b**){  
 minicost();  
 **return**;  
 }  
  
 **for**(**int** j=0; j<=**purch**[i].**piece**; j++){  
 **product**[i] = j;  
 comp(i+1);  
 }  
 }  
 **private void** out(){  
 **button**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **jieguo**.setText(**cost**[**product**[1]][**product**[2]][**product**[3]][**product**[4]][**product**[5]]);  
 }  
 });  
 }  
}

1. 用安卓来写爬取京东数据的代码，通过使用HTTP协议访问网络的方法成功获取了京东的数据，使用OKHttp,安卓自带的库向服务器发出一条HTTP请求，获取服务器返回的数据（代码如下爬取数据代码部分），提取出数据将数据存储在Aapche相应的目录文件里以json格式存储，然后通过JSON格式解析服务器返回来的数据，解析的代码如下（解析数据代码部分）。在爬取的过程遇到了使用JSONObject库来访问电脑本机的服务器地址时访问不成功，爬取价格失败，不能提供出价格。

**爬取数据代码部分：**

**import android.graphics.Paint;**

**import android.graphics.Rect;**

**import android.support.v7.app.AppCompatActivity;**

**import android.os.Bundle;**

**import android.view.View;**

**import android.widget.Button;**

**import android.widget.TextView;**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStream;**

**import java.io.InputStreamReader;**

**import java.net.HttpURLConnection;**

**import java.net.URL;**

**import okhttp3.OkHttpClient;**

**import okhttp3.Request;**

**import okhttp3.Response;**

**public class MainActivity extends AppCompatActivity implements View.OnClickListener {**

**TextView responseText;**

**@Override**

**protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.activity\_main);**

**Button sendRequest=(Button)findViewById(R.id.send\_request);**

**responseText=(TextView)findViewById(R.id.response\_text);**

**sendRequest.setOnClickListener(this);**

**}**

**@Override**

**public void onClick(View v){**

**if(v.getId()==R.id.send\_request){**

**senRequestWithHttpURLConnection();**

**}**

**}**

**private void senRequestWithHttpURLConnection(){**

**new Thread(new Runnable() {**

**@Override**

**public void run() {**

**try{**

**OkHttpClient client=new OkHttpClient();**

**Request request=new Request.Builder()**

**.url("https://item.jd.com/8342912.html")**

**.build();**

**Response response=client.newCall(request).execute();**

**String responseData=response.body().string();**

**showResponse(responseData);**

**}catch (Exception e){**

**e.printStackTrace();**

**}**

**}**

**}).start();**

**}**

**private void showResponse(final String response){**

**runOnUiThread(new Runnable() {**

**@Override**

**public void run() {**

**responseText.setText(response);**

**}**

**});**

**}**

**}**

**解析数据代码部分：**

**import** android.graphics.Paint;  
**import** android.graphics.Rect;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.TextView;  
  
**import** org.json.JSONArray;  
**import** org.json.JSONObject;  
  
**import** java.io.BufferedReader;  
**import** java.io.IOException;  
**import** java.io.InputStream;  
**import** java.io.InputStreamReader;  
**import** java.net.HttpURLConnection;  
**import** java.net.URL;  
  
**import** okhttp3.OkHttpClient;  
**import** okhttp3.Request;  
**import** okhttp3.Response;  
  
**public class** MainActivity **extends** AppCompatActivity **implements** View.OnClickListener {  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 Button sendRequest=(Button)findViewById(R.id.***send\_request***);  
  
 sendRequest.setOnClickListener(**this**);  
  
 }  
 @Override  
 **public void** onClick(View v){  
 **if**(v.getId()==R.id.***send\_request***){  
 senRequestWithHttpURLConnection();  
 }  
 }  
 **private void** senRequestWithHttpURLConnection(){  
 **new** Thread(**new** Runnable() {  
 @Override  
 **public void** run() {  
 **try**{  
 OkHttpClient client=**new** OkHttpClient();  
 Request request=**new** Request.Builder()  
 .url(**"**[**http://10.0.2.2/das.json**](http://10.0.2.2/das.json)**"**)  
 .build();  
 Response response=client.newCall(request).execute();  
 String responseData=response.body().string();  
 parseJSONWITHJSONObject(responseData);  
 }**catch** (Exception e){  
 e.printStackTrace();  
 }  
 }  
 }).start();  
 }  
  
 **private void** parseJSONWITHJSONObject(String jsonData){  
 **try**{  
 JSONArray jsonArray=**new** JSONArray(jsonData);  
 **for** ( **int** i=0;i<jsonArray.length();i++)  
 {  
 JSONObject jSONObject = jsonArray.getJSONObject(i);  
 String id = jSONObject.getString(**"id"**);  
 String name = jSONObject.getString(**"name"**);  
 String version = jSONObject.getString(**"version"**);  
 Log.*d*(**"MainActivity"**,**"id is"**+id);  
 Log.*d*(**"MainActivity"**,**"id name"**+name);  
 Log.*d*(**"MainActivity"**,**"id version"**+version);  
 }  
 }**catch** (Exception e){  
 e.printStackTrace();  
 }  
 }  
}

**解决方法：**

1. 由于对Java和Android studio的学习略有了解，再试了两天用HTML和纯JS不行后打算利用Android Studio进行程序界面设计和代码编写
2. 在运用Android Studio结合JAVA算法代码的时候，遇到需要用户动态提供输入数据的这个难题，输入数据的量是不确定的，所以尝试用数据库存取EditText这个控件获取的用户输入数据，然后再调用数据库的数据进行算法计算，算法过程中结合数据库的调用，由于不懂得封装，虽然说编译没错，但是程序运行还是有错误，调试了多次也实现不了



### 成员总结

用JS的get方式获取请求访问京东页面代码，是否访问成功没有得到验证，无法得知，因没有JS网页制作的经验，获取价格的代码写不出，导致程序中断，通过老师的讲解从HTML网页制作的学习中领悟到了一个项目的完成需要个方面的技术分工完成，从而也得知到我们以后要往那方面走。在学习的过程中，从老师要求我们完成的算法的功能到一步步该怎么做，而不是简简单单的课本知识，而是把课本的知识用到实践上，也让我们了解到为什么要学代码课程，为什么要学算法课程，是为了让这些课程结合在一起在实践上完成一个项目。

1.我们需要做UI界面，实现把商品和数量输入后结合用户的需求输入相应的数据向服务器输入数据。

2.用JS代码写出爬取商品价格（我们爬取不出数据）。

3.将算法写入实现功能。