全国高校大数据应用创新大赛

【大数据技术技能赛】

算法说明书

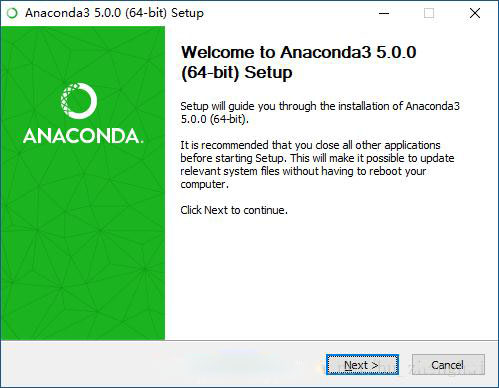
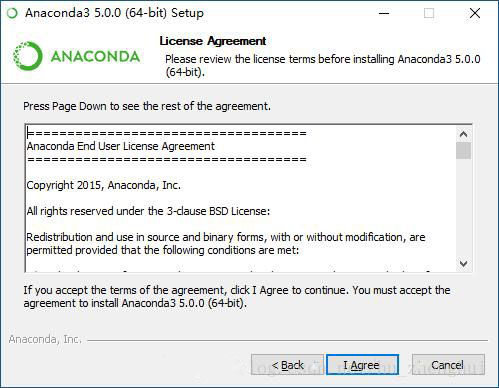
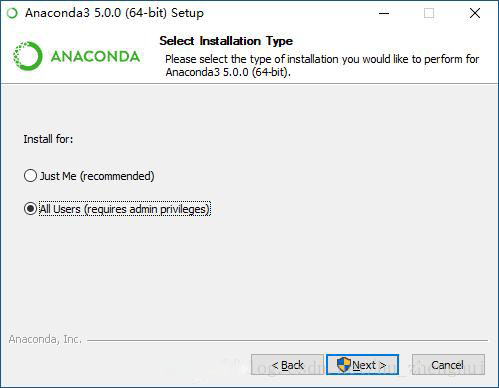
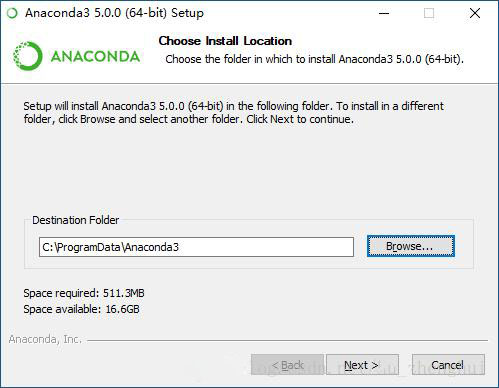
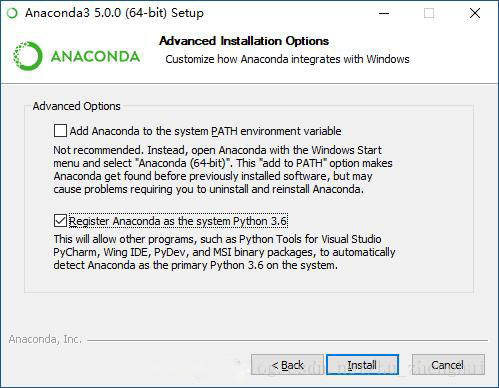
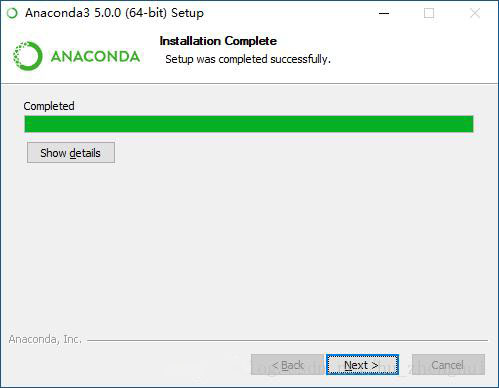
团队名称： 河北机电职业技术学院一队

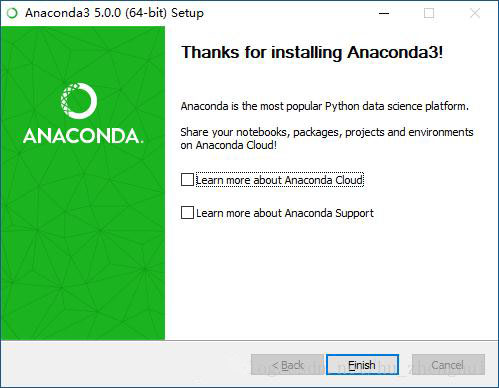
学校学院： 河北机电职业技术学院

填报日期： 2018年6月20

## 开发环境介绍

Window 平台用Anaconda3-5.0集成安装 Python

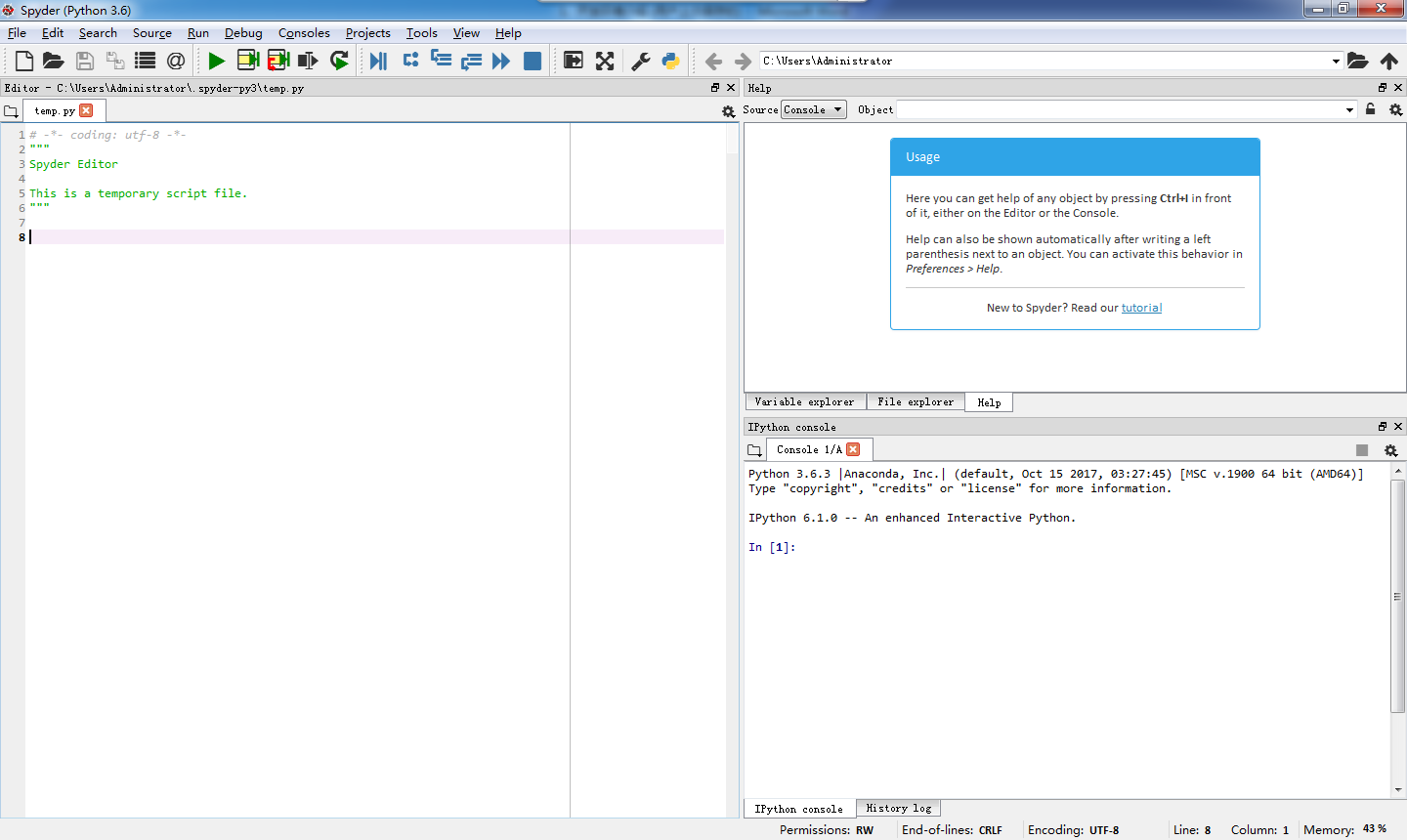
1. 在 “Welcome to Anaconda3 5.0.0 (64-bit) Setup” 界面单击 “Next” 按钮   
2. 在 “License Agreement” 界面单击 “I Agree” 按钮   
   
3. 在 “Select Installation Type” 界面的 “Install for” 中选择 “All Users(requires admin privileges)” 单选按钮   
   如果选择 “Just Me(recommended)” 将安装在 %USERPROFILE% 路径中，如果该路径包含非英文字符，将影响程序执行。   
   单击 “Next” 按钮   
   
4. 在 “Choose Install Location” 界面中查看“Destination Folder”，可能为 C:\ProgramData\Anaconda3，注意检查该路径不能包含非英文字符   
   单击 “Next” 按钮   
   
5. 在 “Advanced Installation Options” 界面中选中 “Register Anaconda as my default Python 3.6” 复选框。 单击 “Install” 按钮   
   
6. 等待安装完成   
   在 “Installation Complete” 界面中单击 “Next” 按钮。   
   
7. 在 “Thanks for installing Anaconda3” 界面中单击 “Finish” 按钮。



1. 打开Spyder程序



1. 编写程序



## 算法介绍

## 

## 源代码展示

# -\*- coding: utf-8 -\*-  
"""  
Created on Sat Jun 2 16:59:14 2018  
  
@author: 额日敦  
"""  
*import* numpy *as* np  
  
# 调用numpy库  
number = np.loadtxt("preliminary-testing.csv", dtype=np.str, delimiter=",")  
# 读取数据文件，以str类型接收（以逗号分隔）  
fw = open('dsjyycxds\_preliminary.txt', 'w')  
# 写入提交文件  
*for* num *in* range(0, len(number)): # 执行 代码 次数  
 even = np.array(number[num]) # 接收要判断的数据 (调用numpy的array函数)  
 color = np.array(even[0:10:2]) # 把数据以特征值分成两种(颜色和数值)  
 value = np.array(even[1:11:2])  
 eight = (value == 'J') # 如果数据有牌值为 J,Q,K 的牌，转化成对应值  
 value[eight] = '11'  
 nine = (value == 'Q')  
 value[nine] = '12'  
 ten = (value == 'K')  
 value[ten] = '13'  
 value = value.astype(int)

# 接收的数值部分转化成int类型，便于后续计算  
 summ = np.sum([value])

# 计算数值部分的总和 (调用numpy的sum函数)  
 Norepeat = len(value) - len(set(value))

# 计算数值不重复的个数  
 value.sort()

# 把数据的数值部分，从小到大升序  
 xiangcha = value[4] - value[0]

# 计算最值之间的长度 (用索引值接收到数值后 计算 最大值于最小值之间的差， 并接收)  
 Tag = 1  
 *while* Tag == 1:  
 # 颜色全部相同的 9，8，5，分为一类  
 *if* color[0] == color[1] == color[2] == color[3] == color[4]:

# 判断接收的数据颜色是否全部相同  
 *if* value[0] == 1 *and* summ == 47:

# 先判断第一个数值是否为A，并且数值总和47  
 print(9, file=fw)

# 输出 9 ，并写入文件里  
 Tag = 0  
 *elif* ((value[0] + value[1] + value[3] + value[4]) / 4 == value[2]) *and* xiangcha == 4:  
 # 再判断数值部分是否前后间隔相同，并且间隔为 1  
 print(8, file=fw)  
 Tag = 0  
 *else*: # 否则就只是颜色全部相同  
 print(5, file=fw)  
 Tag = 0  
 # 数值大小全部不同的 4 和 0 分为一类  
 *elif* Norepeat == 0:  
 *if* (((value[0] + value[1] + value[3] + value[4]) / 4 == value[2] *and* xiangcha == 4) *or* (  
 value[0] == 1 *and* summ == 47)):  
 print(4, file=fw)

# 判断（数值部分是否前后间隔相同，并且间隔为 1）或者（第一个数值是否为A，并且数值总和47）  
 Tag = 0  
 *else*:  
 print(0, file=fw)   
 Tag = 0  
 # 数值大小有 1 张数值重复的  
 *elif* Norepeat == 1:  
 print(3, file=fw)  
 Tag = 0  
 # 数值大小有 2 张数值重复的  
 *elif* Norepeat == 2:  
 # 判断是否有 3 张数值一样的  
 *if* (value[0] == value[1] == value[2] *or* value[1] == value[2] == value[3] *or* value[2] == value[3] == value[  
 4]):  
 print(1, file=fw)  
 Tag = 0  
 *else*:   
 print(2, file=fw)  
 Tag = 0  
 # 数值大小有4 张数值重复的  
 *elif* Norepeat == 3:  
 # 判断是否有 4 张数值一样的  
 *if* value[0] == value[1] == value[2] == value[3] *or* value[1] == value[2] == value[3] == value[4]:  
 print(7, file=fw)  
 Tag = 0  
 *else*:   
 print(6, file=fw)  
 Tag = 0  
fw.close()

## 运行结果展示

