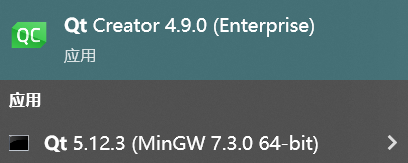
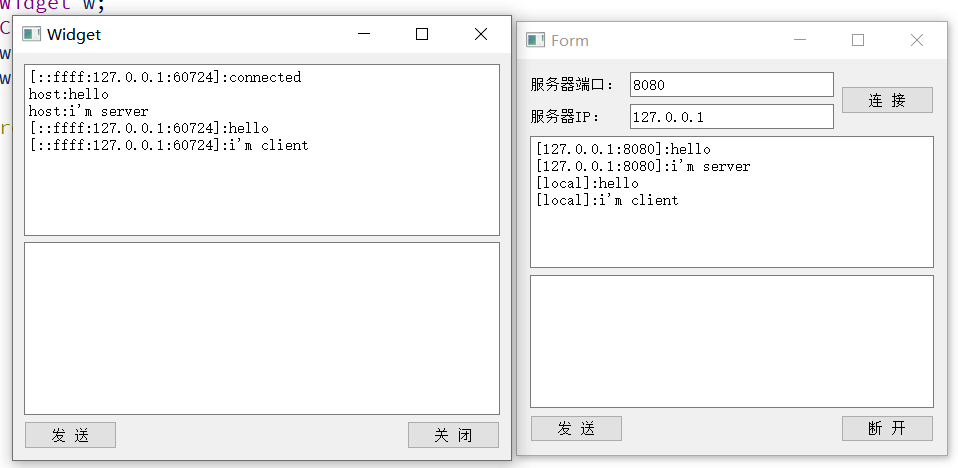
## 运行环境

QT Creator4.9.0

QT 5.12.3（MinGW 7.3.0 64-bit）

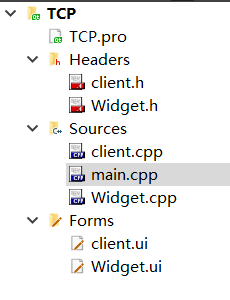


## 运行截图



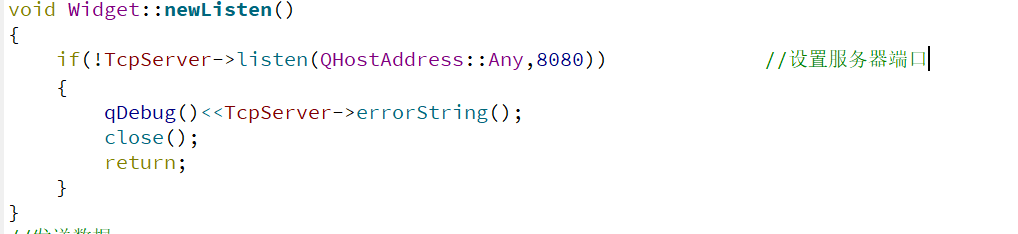
## 操作流程

项目结构如下：

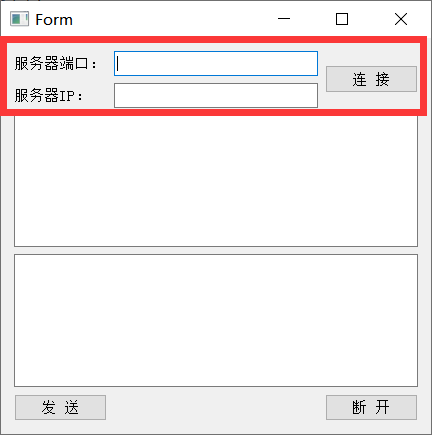


client为客户端，Widget为服务器端，运行main函数开始程序。

在Widget类中设置通信端口



运行main后在客户端输入服务器端ip和端口点击连接即可。

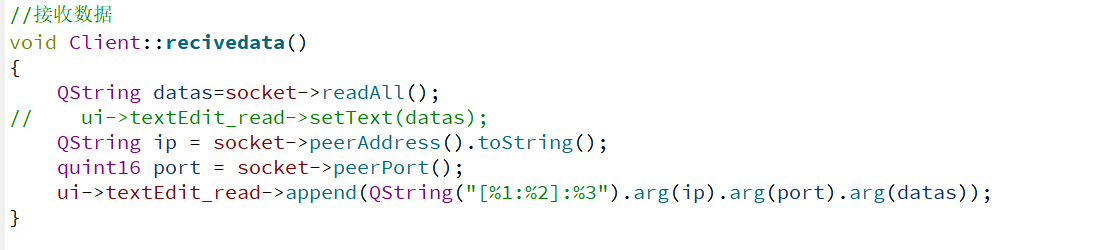


## 相关函数

基于QTcpServer和QTcpSocket实现。

1. client

建立连接后通过socket的readAll方法获取数据，显示在ui界面

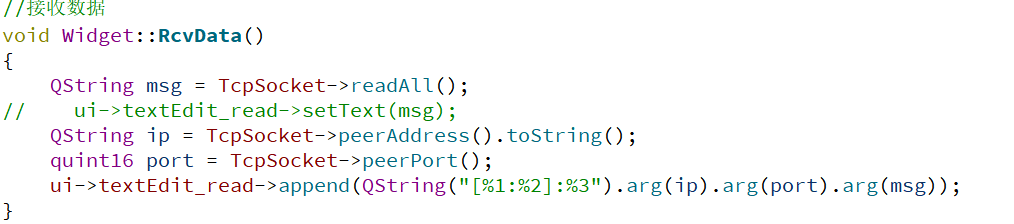


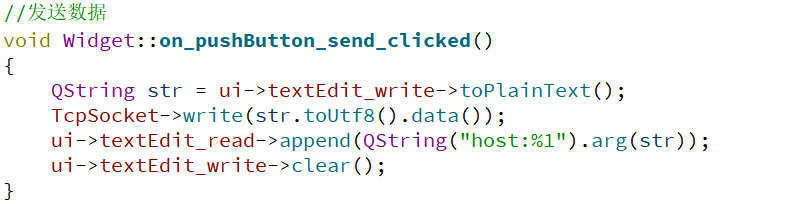
通过socket将数据写回服务器端，并清空输入栏



1. server

通过QTcpsocker类接受数据





Client

客户

端

随机数

数组

intMSG

定时器

每隔

30

s

自动

生

成

1

万

条

随机数

二进制

数组

调用

CreateRandNums

类

中的

ToBinaryCode

()

或者

ToHaffmanCode()

函数

将整形数组

**编码**

为

二进制编码的数组

Server

服务器端

Network

调用

CreateRandNums

类

中的

transform

将

数

组转为字符串通过

socket

发送到服务器端

二进制

字符串

随机数

数组

intMSG

调用

CreateRandNums

类

中

的

BinaryCodeToIntArray

()

或

者

HaffmanCodeToIntArray

()

函数

将

二进制

(

字符串

/

数组

)

**解码**

为

十进制

的

整形

数组

动态

输

出

最大

的

20

个

数

使用

最大堆

求解

动态更新的数据

中的最大的

20

个

数

，

并尝试与其他

类似

算法

对比

二进制字符串

系统流程图，包含新增需求

## 需求说明

参考上图中的系统流程概述，理解系统整体流程和新增需求。程序模板中已在客户端client.cpp中设置定时器每隔30s自动调用onTimeOut函数发送1万个随机数去服务器端，请基于本模板完成以下功能（不强制要求自己实现Haffman树和最大堆类，可以调用STL或者其他开源代码）：

1. 为了模拟二进制数据传输，请自行完成CreateRandNums类中的ToBinaryCode()和ToHaffmanCode()函数，实现在客户端对随机数数组的常规二进制编码或者Haffman编码（对比无压缩和有压缩的数据传输开销），并基于函数TransformBinaryCode()将二进制数组转为字符串进行网络传输；
2. 为了在服务器端进行数据解码，请自行完成CreateRandNums类中的BinaryCodeToIntArray ()和HaffmanCodeToIntArray ()函数，实现在服务器端对接收到的二进制字符串(数组)解码到十进制随机数数组



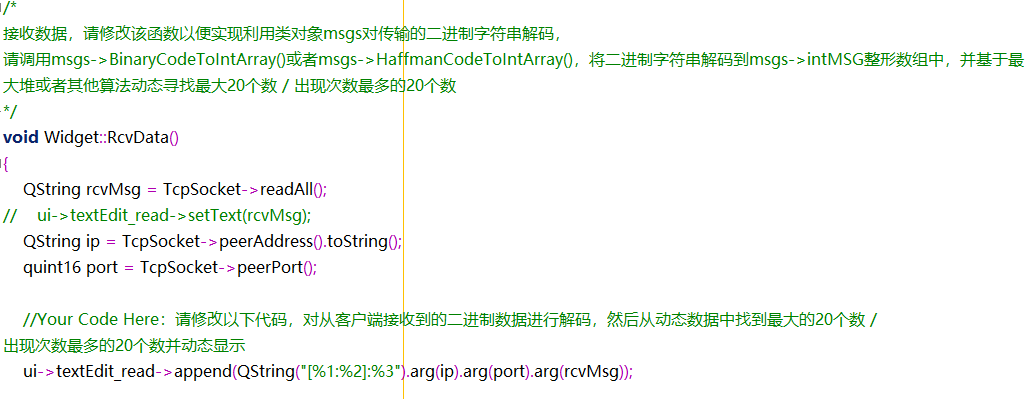
需要完成的代码1：位于CreateRandNums.h中，Client和Server中均需使用该代码实现编码和解码

1. 在客户端，需要修改成传输二进制编码



需要完成的代码2：位于client.cpp中，主流程已提供

1. 在服务器端，请基于最大堆求解从客户端接收到的动态数据中（包括历史数据）的**最大20个数 / 出现次数最多的20个数**，并动态展示，此外，请尝试与其他算法对比，例如最朴素的遍历查找算法等



需要完成的代码3：位于server.cpp中