



寻路山威——山东大学（威海）校园导航APP

计算机科学与技术（中外合作）01班



指导老师：姜斌



答辩人：孙功新

目录

CONTENTS



01 | 绪论

- ✓ 选题背景
- ✓ 研究意义



03 | 功能解析

- ✓ 我的位置
- ✓ 地标全览
- ✓ 两点导航
- ✓ 随机导航



05 | 总结致谢

- ✓ 个人收获



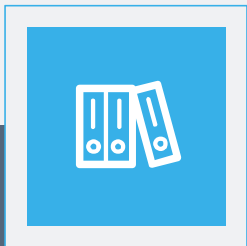
02 | 项目概览

- ✓ 研究思路
- ✓ 研究方法



04 | 项目展望

- ✓ 室内导航
- ✓ 地图扩充



01 | 绪论

- ✓ 选题背景
- ✓ 研究意义



绪论



项目概览



功能解析



项目展望



总结

需求

- 对比腾讯、百度、高德、谷歌等地图，对于校内地标标注的不完善
- 校内的小路在地图软件中没有体现





绪论



项目概览



功能解析



项目展望



总结

和课程的联系

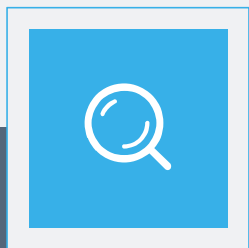
导航软件是以**最短路径算法为核心**的系统，与数据结构这门课联系紧密。是**数据结构知识应用**的完美体现。



对自我的提升

- 接触**移动端开发**
- 自己**独立完成**app开发从无到有
- 通过文献阅读，对**最短路径算法**有更深刻的认识





02 | 项目概览

- ✓ 研究思路
- ✓ 研究方法



寻路山威

Version: 1.0

平台

用户可直接将“寻路山威.apk”下载到Android设备使用



外观

配色山大红+白
appicon图

整体设计

- 极简设计 用户友好
- 无需登录，直接使用
- 界面简洁明了
- 功能易上手



绪论



项目概览



功能解析



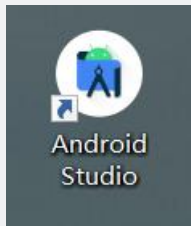
项目展望

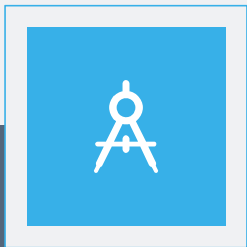


总结

语言

Android Studio





03 | 主要功能

- ✓ 我的位置
- ✓ 地标全览
- ✓ 两点导航
- ✓ 随机导航

功能概览



绪论



项目概览



功能解析



项目展望

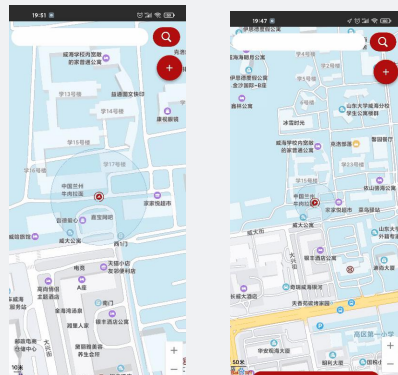


总结

我的位置



两点导航



地标全览



随机导航



我的位置



绪论

项目概览

功能解析

项目展望

总结



- Baidu Map API
- 打开GPS
- 设置Toast提醒



```
/**
 * 开始定位函数
 * @throws Exception
 */
private void setMapLocationData() throws Exception{
    // 开始定位函数
    mBaiduMap.setMyLocationEnabled(true);
    // 构造定位数据
    MyLocationData locData = new MyLocationData.Builder()
        .accuracy(50)
        // 此处设置开发获取到的方向信息，默认0-360
        .direction(azimuth)
        .latitude(mBaiduMap.getLocationData().latitude)
        .longitude(mBaiduMap.getLocationData().longitude).build();
    // 设置定位数据
    mBaiduMap.setMyLocationData(locData);
    throw new Exception();
}
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    // 初始化地图
    ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
    // 初始化地图
    mBaiduMap = (BaiduMap) findViewById(R.id.bmapview);
    mBaiduMap.setMapType(BaiduMap.MapType.NORMAL);
    mBaiduMap.setMyLocationEnabled(true);
    // 初始化定位
    mLocationListener = new MyLocationListener() {
        @Override
        public void onReceiveLocation(BaiduMap bmap, MyLocationData locData) {
            Toast.makeText(this, "经纬度: " + locData.getLatitude() + ", " + locData.getLongitude(),
            Toast.LENGTH_SHORT).show();
        }
    };
    mBaiduMap.setLocationListener(mLocationListener);
    // 初始化定位
    mLocationListener = new MyLocationListener() {
        @Override
        public void onReceiveLocation(BaiduMap bmap, MyLocationData locData) {
            Toast.makeText(this, "经纬度: " + locData.getLatitude() + ", " + locData.getLongitude(),
            Toast.LENGTH_SHORT).show();
        }
    };
    mBaiduMap.setLocationListener(mLocationListener);
}
```



```
private void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    // 初始化地图
    ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
    // 初始化地图
    mBaiduMap = (BaiduMap) findViewById(R.id.bmapview);
    mBaiduMap.setMapType(BaiduMap.MapType.NORMAL);
    mBaiduMap.setMyLocationEnabled(true);
    // 初始化定位
    mLocationListener = new MyLocationListener() {
        @Override
        public void onReceiveLocation(BaiduMap bmap, MyLocationData locData) {
            Toast.makeText(this, "经纬度: " + locData.getLatitude() + ", " + locData.getLongitude(),
            Toast.LENGTH_SHORT).show();
        }
    };
    mBaiduMap.setLocationListener(mLocationListener);
    // 初始化定位
    mLocationListener = new MyLocationListener() {
        @Override
        public void onReceiveLocation(BaiduMap bmap, MyLocationData locData) {
            Toast.makeText(this, "经纬度: " + locData.getLatitude() + ", " + locData.getLongitude(),
            Toast.LENGTH_SHORT).show();
        }
    };
    mBaiduMap.setLocationListener(mLocationListener);
}
```

地标全览



本功能选取了山东大学（威海）内部较为著名的建筑物进行了标注，并且附有图片及文字描述，可以帮助小白很快的了解到学校风光。

①地标选取：

- 教学场所（各个学院楼）
- 体育场所（体育馆、操场、游泳馆等）
- 饮食场所（四个餐厅）
- 生活场所（外卖取餐点、浴池、快递点）
- 其他（如玛珈山等）

②地标描述：

图片（更加直观）+文字（地点介绍）



两点导航



本功能名为“两点导航”，提供点对点的寻路服务。用户可以手动选择两栋建筑物并选择游览方式（步行/骑行），系统自动计算出相应的最短路径。

输入图

- 顶点经纬度信息输入
- 路径长度输入

选取顶点

- 地标全览中的地标；
- 补充地点（具体公寓楼、建筑的多个门等）

04

03

02

01

最短路算法

多种算法对比

选取边

- 步行
- 骑行

两点导航

为您规划山东大学（威海）最佳路线

出发点



目的地



步行寻路



骑行寻路



取消



绪论



项目概览



功能解析



项目展望



总结

29个地标点+39个辅助点

体育、干饭、自习、住宿



绪论



项目概览



功能解析



项目展望



总结

选择出发点

南门

艺术学院 (音乐舞蹈楼)

知行会堂

知行楼

海洋学院

商学院

外籍专家公寓 (快递收发点)

外卖点

浴池

荟园餐厅

馨园餐厅

泰园餐厅

雀园餐厅 (大学生活动中心)

法学院

图书馆

选择出发点

图书馆

数学与统计学院

玲珑学堂

电子楼 (联合理学院)

文学楼 (文化传播学院)

机电与信息工程学院

科学实验楼

闻外楼 (翻译学院、马列部)

田径场

东北亚学院

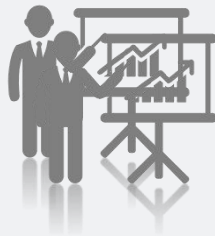
体育馆

游泳馆

闻天楼 (空间科学与物理学院)

网络楼

天文台



两点导航-边选取



绪论



项目概览



功能解析



项目展望



总结



两点导航

为您规划山东大学（威海）最佳路线

出发点



目的地



步行寻路

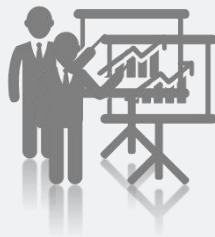


骑行寻路



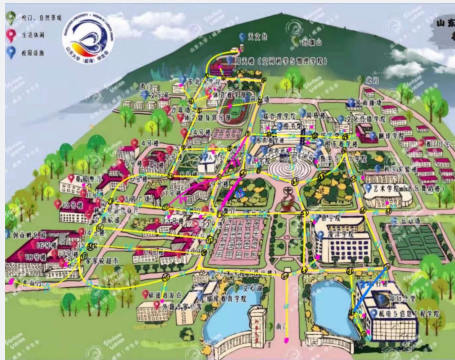
取消

有些路只能步行！



两点导航-输入图

纸上初步建模：（83条边）



输入经纬度等构图：

```
// 主节点(节点介绍中的节点)
new Point( latitude: 37.532649, longitude: 122.067089, name: "南门", index: 0), //0
new Point( latitude: 37.536654, longitude: 122.069487, name: "艺术学院(音乐舞蹈楼)", index: 1), //1
new Point( latitude: 37.533603, longitude: 122.068323, name: "知行会堂", index: 2), //2
new Point( latitude: 37.534088, longitude: 122.068973, name: "知行楼", index: 3), //3
new Point( latitude: 37.534817, longitude: 122.067462, name: "海洋学院", index: 4), //4
new Point( latitude: 37.535754, longitude: 122.067528, name: "商学院", index: 5), //5
new Point( latitude: 37.533389, longitude: 122.06406, name: "外籍专家公寓(快速收发点)", index: 6), //6

new Point( latitude: 37.53347, longitude: 122.06009, name: "外卖点", index: 7), //7
new Point( latitude: 37.534052, longitude: 122.062932, name: "浴池", index: 8), //8
new Point( latitude: 37.53545, longitude: 122.063721, name: "茶园餐厅", index: 9), //9
new Point( latitude: 37.535502, longitude: 122.064274, name: "茶园餐厅", index: 10), //10
new Point( latitude: 37.535994, longitude: 122.062419, name: "茶园餐厅", index: 11), //11
new Point( latitude: 37.537339, longitude: 122.064141, name: "茶园餐厅(大学生活动中心)", index: 12), //12

new Point( latitude: 37.537552, longitude: 122.065818, name: "法学院", index: 13), //13
new Point( latitude: 37.538067, longitude: 122.066675, name: "图书馆", index: 14), //14
new Point( latitude: 37.53747, longitude: 122.067496, name: "数学与统计学院", index: 15), //15
new Point( latitude: 37.538369, longitude: 122.066743, name: "珠宝学院", index: 16), //16
new Point( latitude: 37.538783, longitude: 122.066193, name: "电子楼(联合学院)", index: 17), //17
new Point( latitude: 37.538727, longitude: 122.068747, name: "文学楼(文化传播学院)", index: 18), //18
new Point( latitude: 37.538076, longitude: 122.068246, name: "机电与信息工程学院", index: 19), //19
new Point( latitude: 37.537189, longitude: 122.069089, name: "科学实验楼", index: 20), //20

new Point( latitude: 37.538523, longitude: 122.0694, name: "南外楼(翻译学院、马列部)", index: 21), //21
new Point( latitude: 37.538924, longitude: 122.064647, name: "田径场", index: 22), //22
new Point( latitude: 37.539924, longitude: 122.062113, name: "东北亚学院", index: 23), //23
new Point( latitude: 37.539689, longitude: 122.062753, name: "体育馆", index: 24), //24
new Point( latitude: 37.539113, longitude: 122.063027, name: "图书馆", index: 25), //25
new Point( latitude: 37.540902, longitude: 122.063819, name: "闻天楼(空间科学与物理学院)", index: 26), //26
new Point( latitude: 37.538743, longitude: 122.067219, name: "综合楼", index: 27), //27
```





绪论



项目概览



功能解析



项目展望



总结

分析图

图中有83条边, 68个顶点。可得两个结论如下:

(1) 因为 $83 < 68 * \log 268$, 故为稀疏图;

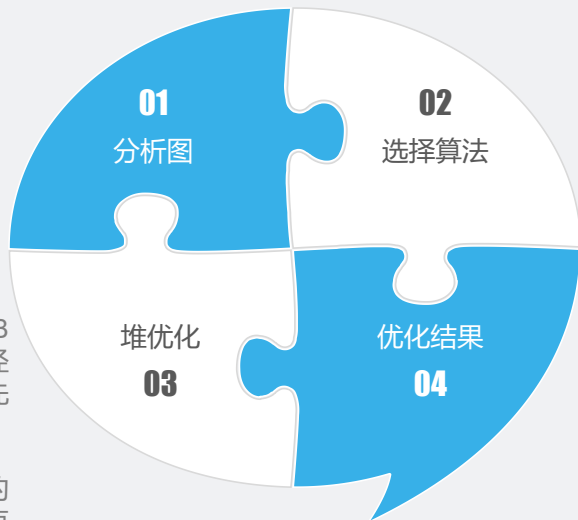
(2) 因为路径权值 >0 , 故边全部非负。

堆优化

内层循环的主要功能是找出点集B (未求出最短路) 中源点到其路径最短的点, 这个地方可以使用优先队列进行优化。

选取二叉最小堆 (一种经过排序的完全二叉树, 其中任一非终端节点的数据值均不大于其左子节点和右子节点的值) 进行优化。

由于二叉堆的性质, 根节点的值最小。所以获取优先级最高的值只需要将根节点返回即可 $O(1)$, 然后调整堆。该操作的时间复杂度为 $O(\lg n)$ 。



选择算法

Floyd: $O(n^3)$ 暴力枚举

✓Dijkstra: $O(n^2)$ 贪心算法 没有负边

优化结果

一开始将所有的点 (除了源点) 放至二叉堆中, 随后查找节点i的时候, 直接调用getTop()方法, 时间复杂度为 $O(\lg n)$ 。

由于对每个节点调用一次, 因此总共需要n次这个操作。

在获取到点i之后, 通过点i进行松弛, 松弛之后需要更新该节点, 因此需要调用update()方法。需要松弛的其实并非所有节点, 而只是点i的临接点。因此, update操作需要调用E (图中边的数量) 次。

因此, 使用二叉堆优化后的Dijkstra算法的复杂度为 $O((E+n)\lg n)$, 因此该优化适合于稀疏图。

下图所示为系统为从“浴池”到“电子楼”生成的三种路径
(从左至右依次为: 步行、电动车、自行车)



绪论

项目概览

功能解析

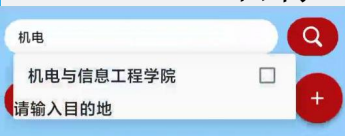
项目展望

总结

相较于两点导航，随机导航更符合非游客用户习惯。用户只需输入目的建筑物名（前提是建筑物已经存于系统图中），系统即自动规划从当前位置到目的地的最短路径。

AutoComplete

自动补全地标
名称



机电

机电与信息工程学院

请输入目的地

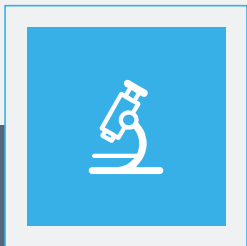
最短步行路径导航

将此刻位置先通过Link(G)函数用经纬度计算找到距离我的位置最近的地标，再调用两点导航。

选择热门地标

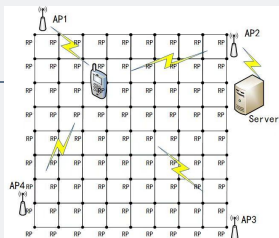
使用的仍然是前文所述地标

```
private void link(Vector<Node>[] G){
    Point temp=null;
    //先将min设置为0
    double minn=Double.MAX_VALUE;
    Point startPoint=points[startArrayIndex];
    for(int i=1;i<points.length;i++){
        //计算startPoint到其他点的距离
        double dis=DistanceUtil.getDistance(new LatLng(startPoint.latitude,
        //找startpoint到其他某点的最小距离
        //temp存点; minn存最小distance
        if(dis<minn){
            minn=dis;
            temp=points[i];
        }
    }
    //存为Node (彼此的距离为10)
    G[startArrayIndex].add(new Node(getArrayIndex(temp.index),10));
    G[getArrayIndex(temp.index)].add(new Node(startArrayIndex,10));
}
```



04 | 项目展望

- ✓ 室内导航
- ✓ 地点扩充



室内导航

考虑到以“图西教学楼”为代表的一众**教学楼内部地形较为复杂**，而且**食堂窗口**的具体地址也是同学们关心的热点之一。所以在设计初期考虑为食堂及部分教学楼提供室内导航。

但是经过调研发现室内导航实现难度很大。原因如下：

①一般室内使用WiFi指纹定位法（不是所有的地方有大量WiFi覆盖。每个位置至少得同时扫描到两三个较强的WiFi才能定位，WiFi数量越多理论上定位精度越高。）

②硬件设施不够，普及为时尚早：室内定位与室外定位的差别在于室外有GPS这种统一的基础性设施和服务，室内定位没有。

哈啰单车、校易行接口

两点导航时，可以选择：自行车/电动车。
自动跳转到哈啰单车/校易行开始旅程。

- 增加更多学校地点（完善公寓楼群）；
- 补充建筑物不同入口

地图扩充



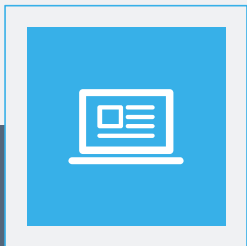
绪论

项目概览

功能解析

项目展望

总结



05 | 总结

- ✓ 问题评估
- ✓ 相关对策
- ✓ 研究总结
- ✓ 成绩与思考



绪论



项目概览



功能解析



项目展望



总结



收获一

get到了新的技能和知识。通过这次的项目，我第一次接触了百度地图API的调用以及基于Android平台的app搭建，随着项目的推进，这些技能被我慢慢熟悉和掌握。

收获二

扩展了已有知识，对数据结构的认识更加深刻。因为本项目的核心算法在于最短路径算法，所以在实现期间，我搜集了很多资料，对比了很多算法，想要找到最佳的算法来提高APP的性能。在这个过程中非常令我震惊的一点是：提高算法性能的关键竟然不在于算法本身逻辑，而在于底层的数据存储结构。这个知识只有在实践中才能充分领悟到。

收获三

APP的搭建从无到有，这种自我成就感也是本次项目带给我的最大收获之一。虽然结果与预期还有一定差距，但我还是在这次项目中汲取了力量，从而能够优化APP，继续前行。



感谢您的观看！

计算机科学与技术（中外合作） 孙功新



指导老师：姜斌



答辩人：孙功新