**第46届世界技能大赛移动应用开发项目中国集训队**

**集训日志（选手） 朱姚飞**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **10-13** | **指导老师** | **刘雄华** |
| **训练任务** | **数据结构与算法(图)** | | | |
| **训练内容：**  **图 以广度优先进行遍历**  **训练要求：**  **过程记录：**  List<UnitBox> scan(List<List<UnitBox>> map, UnitBox scanUnit) {  List<UnitBox> result = [];  List<int> rawOffset = [-1, 0, 1];  rawOffset.forEach((a) {  rawOffset.forEach((b) {  try {  final getElem = map[scanUnit.y + a][scanUnit.x + b];  if ((a.abs() != 1 || b.abs() != 1) && (a != 0 || b != 0) && getElem.scanOk == false) {  getElem.scanOk = true;  result.add(getElem);  }  } catch (e) {}  });  });  return result; }  List<UnitBox>? bfs(List<List<UnitBox>> map, UnitBox startUnit, UnitBox targetUnit) {  /// clear state  map.expand((element) => element).forEach((element) => {element.scanOk = false});  List<UnitBox> scanNodes = [startUnit];  Map<UnitBox, UnitBox> pathMap = new Map();  while (scanNodes.length > 0) {  final \_l = List<UnitBox>.from(scanNodes);  scanNodes.clear();  for (var elemNode in \_l) {  final scanGetNodeAgg = scan(map, elemNode);  for (final nextNode in List.from(scanGetNodeAgg)) {  if (nextNode == targetUnit) {  final List<UnitBox> resultList = [nextNode];  UnitBox? endLinkNode = elemNode;  while (endLinkNode != null) {  resultList.add(endLinkNode);  endLinkNode = pathMap[endLinkNode];  }  return resultList;  }  if (nextNode.type != null) {  scanGetNodeAgg.remove(nextNode);  }  }  scanGetNodeAgg.forEach((mapKey) {  pathMap[mapKey] = elemNode;  });  scanNodes.addAll(scanGetNodeAgg);  }  }  return null; }  **分析总结：**  **图 BFS算法广度优先搜索算法** | | | | |
|  | | | | |

**填写人：朱姚飞**