Machine Translated by Google

COMP7107 复杂数据类型的管理

作业2

点数据

截止日期:2024年3月15日下午5:00

这项任务的目标是开发索引和搜索空间数据的技术。

第 1 部分（30%,指数开发）

从 Moodle 下载文件Beijing_restaurants.txt。该文件包含 51970 个点的坐标,其中
是北京的餐馆地点。第一行有点数。从第二行开始,每行包含餐厅的坐标（x 和 y）。您需要编写一个程序来构造一个简单的空间网格索引。网格将点覆盖的区域划分为 10*10 = 100 个大小相等的矩形（单元格）。

要构建网格,您应该首先将文件中的点读取到内存中的数据结构中,然后
确定每个维度的最小值和最大值。对于每一点,你应该给出一个
标识符（唯一的整数)应与包含该点的文件中的行号相同。
例如,第一行中的点（39.856138,116.42394)的标识符应为1,点（39.813336,116.486149)的标识符应为2等。然后,将每个维度的值范围划分为10个大小相等的区间。下一步是根据包含点的单元格对点进行排序。这意味着单元格 (0,0) 中的所有点应位于单元格 (0,1) 中的所有点之前,等等。将排序后的点写入文本文件 grid.grd,其中每行的形式为 <标识符 x 坐标 y -坐标>。另外,创建另一个文件

grid.dir（目录）,包含以下内容。 grid.dir 的第一行应该具有每个维度的最小和最大值。然后,每个非空单元格应该有一行,其中包括单元格在网格中的坐标（即（0,0）、(0,1)等）、文件 grid.grd 中的位置（在字符术语）,其中包含该单元格中第一个点的第一行开始以及单元格中的点数。这样,如果我们有一个单元格的坐标,我们可以使用文件 grid.dir 中的数据来定位并加载该单元格的所有点。

在下图中,您可以看到两个文件的前 11 行的快照以及网格的图示示例。在图示示例中,观察如何确定将每个轴的范围划分为 10 个间隔的值。每个点都分配给包含它的单元格。如果某个值恰好落在网格的一条线上,则该值将分配给该线后面的单元格。例如,x 坐标等于 39.68 + (40.18-39.68)/10 且 y 值为 116.072 的点落在单元格 (1,0) 中。

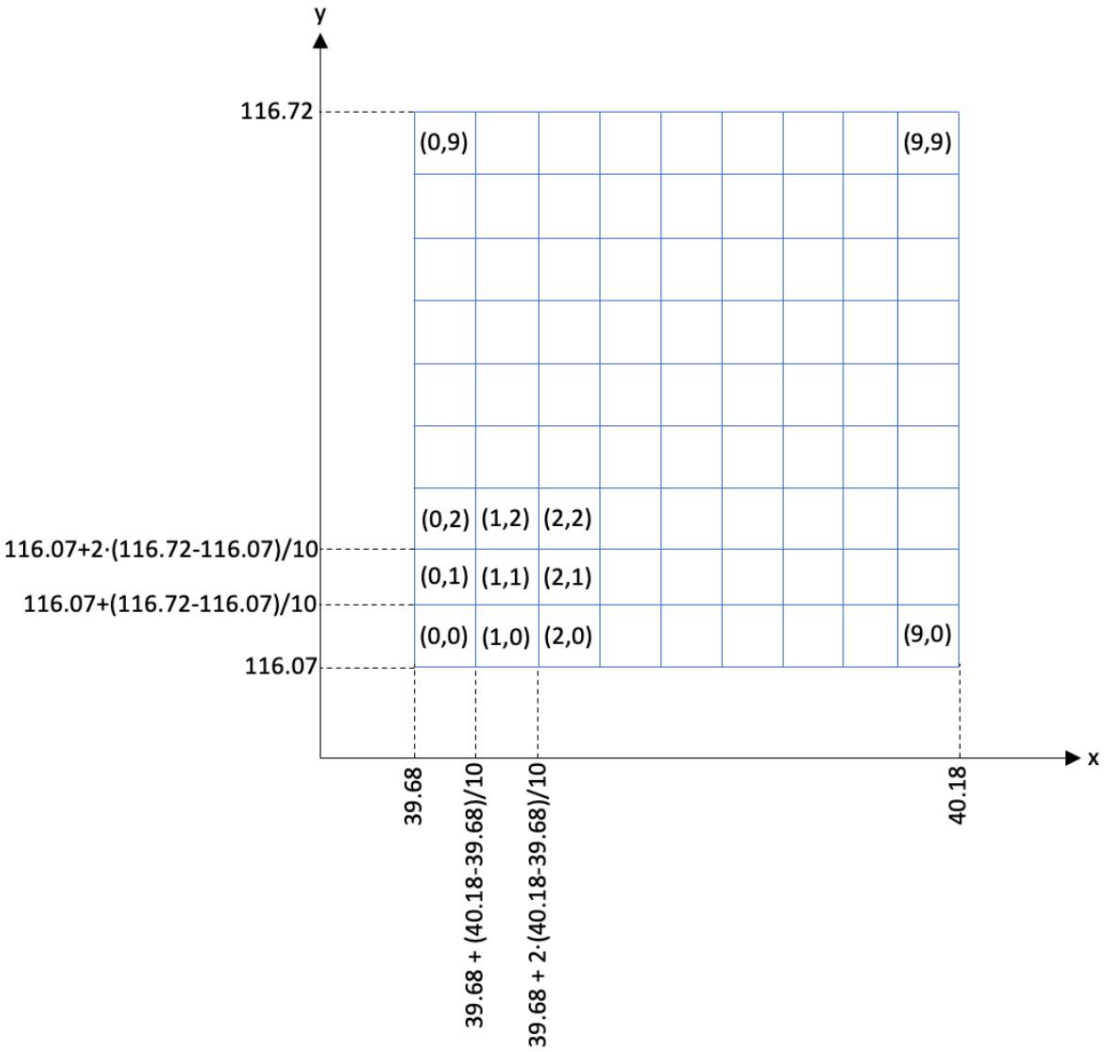
grid.dir 文件的第一行

39.680090 40.179911 116.070466 116.719976
0 0 0 108
0 1 2894 179
0 2 7688 20
0 3 8226 117
0 4 11368 356
0 5 20915 91
0 6 23352 58
0 7 24907 18
1 0 25393 83

grid.grd 文件的第一行

56 39.729270 116.119278
573 39.729398 116.128704
1253 39.723127 116.121828
1372 39.729585 116.127883
1395 39.729571 116.128738
2692 39.706018 116.123828
2846 39.727021 116.121720
3427 39.726623 116.120578
3804 39.723701 116.123683
4146 39.728675 116.135041

Machine Translated by Google



通过交叉检查三个文件的内容来验证程序的正确性:grid.grd、grid.dir
和Beijing_restaurants.txt。

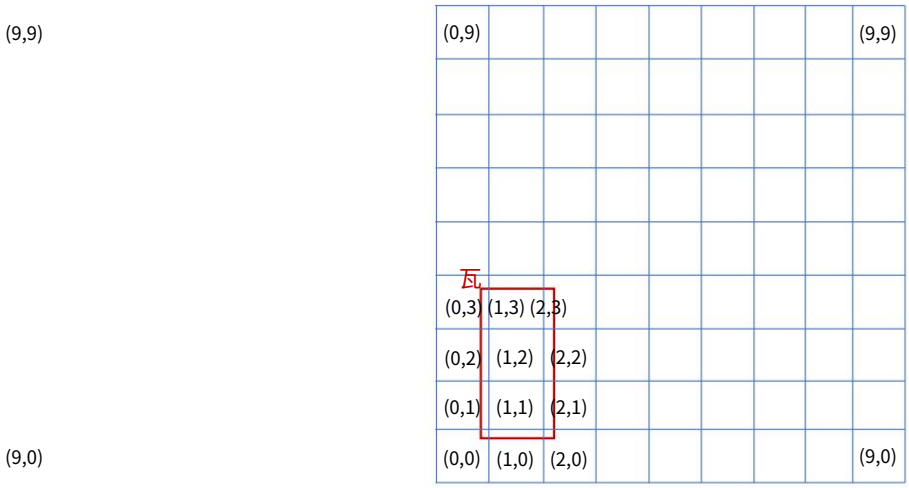
第 2 部分（30%,范围选择查询）

您需要编写一个程序,该程序将使用您在第 1 部分中构建的网格来评估窗口
选择查询。您的程序应将每个维度的窗口下限和上限作为命令行参数,即四个值<x_low> <x_high> <y_low> <y_high>。它应该计算
和
打印窗口中包含的点。

首先,程序应该读取文件 grid.dir 的全部内容并将它们存储在数据结构中。
该信息将用于查询评估。然后它应该读取与窗口相交的每个单元格的所有数据。如果单元格完全被窗口覆盖,则应报告单元格中的所有点而不进行任何比较。如果单元格仅被窗口部分覆盖,那么我们需要验证单元格中的每个点是否包含在窗口中。在下面的图示示例中，对于窗口

查询W,我们报告单元格 (1,2) 和 (1,1) 中的所有点而不检查它们的坐标,而对于单元格 (0,3),(1,3),(2,3),(0,2),(2,2),(0,1),(2,1),(0,0),(1,0) 和 (2,0) 我们必须将点的坐标与窗口来确认它们是否在W内部。

Machine Translated by Google



通过将结果与直接评估窗口查询时所获得的结果进行比较来检查正确性
关于文件Beijing_restaurants.txt 的所有点。为了进行检查,您应该计算并返回每个查询与W相交的单元格数量。

第 3 部分（40%,最近邻查询）

开发一个程序,使用您在第 1 部分中构建的网格索引来增量评估最近邻居查询。您的程序应将数字 k 和查询点 q 的坐标作为命令行参数。它应该输出 k 个最近的餐馆。最初,程序读取整个文件 grid.dir

并将其内容存储在主存数据结构中,该内容应用于查询评估。这
最近邻搜索函数应作为迭代器实现（例如作为生成器函数）,以便在每个函数调用中返回下一个最近邻。程序应该调用该函数 k 次

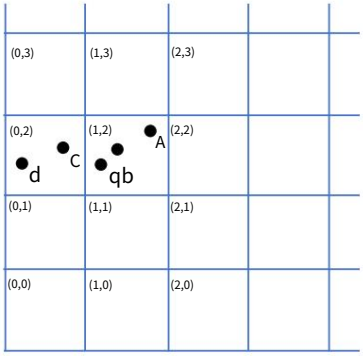
以距离递增的顺序查找并打印 q 的 k 个最近邻。

您的函数应该使用优先级队列,根据单元格和点到 q 的距离来管理它们。
最初,该函数应将距离 q 最近的单元添加到队列中（例如,包含 q 的单元）。然后,该函数应该对最接近 q 的元素（即堆的顶部元素）进行去堆操作。如果此元素是一个单元格,则应从文件 grid.grd 中读取单元格内的所有点（您可以使用 grid.dir 中的数据来定位它们）,并将所有这些点添加到堆中。此外,之前从未添加到堆中的脱堆单元的相邻单元也应该添加到堆中。如果当前释放的元素是一个点,则将其作为 q 的下一个最近邻居返回。

考虑下面的示例,它描述了一些单元格、其中的一些点（a、b、c、d)和一个查询点
q.假设我们正在运行您必须实现的功能。由于 q 位于单元格 (1,2) 内部,因此该单元格首先作为唯一的堆内容添加到堆上。然后,单元格 (1,2) 被去堆,我们将单元格 (1,2) 和 (ii) 单元格 (0,3)、(1,3) 中包含的点 a 和 b 添加到堆 (i) 上)、(2,3)、(0,2)、(2,2)、(0,1)、(1,1)、(2,1) 是 (1,2) 的邻居。距离 q 下一个最近的堆元素是单元格 (0,2),它已脱堆,我们添加单元格 (0,2) 内的堆点 c 和 d。但是,我们不添加单元格的任何相邻单元格

(0,2)到堆中,因为之前已经全部添加到堆中了。下一个被释放的元素
是b点。由于 b 是一个点,因此将其报告为下一个最近邻点。当再次调用该函数时,
通过对距 q 的下一个最接近的元素（即单元 (1,1))进行去堆操作,从停止点继续。全部
将单元格 (1,1) 及其相邻单元格 (0,0)、(1,0)、(2,0) 的内容添加到堆中,并且算法继续对单元格 (0,1) 进行解堆,其从文件 grid.grd 中读取内容并将其添加到堆中。

Machine Translated by Google



您必须编写一个函数 Mindist,它计算点 (q) 和单元格之间的最近欧几里得距离。这个距离可以由每个轴上的距离组成,正如我们在课堂上解释的那样:它是每个维度上的距离平方和的平方根。

除了距离 q 最近的 k 个点之外,您的程序还应该打印您的函数读取其内容的单元格。

可交付成果:使用 Moodle 提交作业所有三个部分的代码。您还应该提交一份
PDF 文件中的报告,它对您的代码进行了高级解释,并包括运行程序的任何说明。您可以使用您选择的编程语言,但您的程序应该独立于操作系统。
请于 2024 年 3 月 15 日下午 5:00 或之前向 Moodle 提交包含所有请求的程序和文档的单个ZIP文件。确保所有内容可读。请不要提交任何数据文件。如果您在此作业中遇到任何困难,请随时在Moodle 论坛上发布您的问题或联系课程的助教。我们很乐意提供帮助。