Data Structures:
Active Learning 1: Introduction

YoungWoon Cha

Spring 2022

Active Learning

- Self-directed team project.
  - Self-study the topic
  - Deliver your thoughts & knowledge
  - Develop your communication & leadership skills

- In this Class,
  - You will participate in two (active learning) group projects
  - Team arrangement session will be held soon.

Active Learning 1

Active Learning 1

- There are 2 parts to AL1
  - Problem1: kd-tree programming
  - Problem2: region quad tree programming
  - ** The purpose of this exercise is to
    - help you understand kd tree, quad tree in full detail
    - train you to review code written by others.
    - Self-directed learning of approximate search algorithms

- Submit a team report to CyberCampus.
  - Source code and test result screen shots

Problem 1: K-D Tree Programming

- Use the following source code for kd-tree
  - https://rosettacode.org/wiki/K-d_tree

- Using the source code provided as basis, implement the following algorithms.
  - P1-1. point_search for a user-specified point
  - P1-2. range_search (find all points contained within a specified bounding rectangle)
  - P1-3. nearest_neighbor_search (given a point, find one or more nearest neighbor points)

Problem 1: K-D Tree Specification

- Assume a 10 x 10 grid.
- Insert the following points in order:
  - (2,3), (5,4), (3,4), (9,6), (4,7), (8,1), (7,2)
- Test the point_search function, by searching for the following points and displaying the results.
  - (5,4), (4,7), (10,5)
- To test the range_search function, specify a rectangle with (left x=6, left y=3), width 3,height 4.
- Test the nearest_neighbor_search function twice: first with input (5,4); then with input (4,7)

Problem 2: Quad Tree Programming

- Reference the source codes for region quad tree
  - https://iq.opengenus.org/quadtree/
  - https://www.geeksforgeeks.org/quad-tree/

- Using the source code provided as basis, implement the following algorithms.
  - P2-1. build_tree (build a quad tree, given N points)
    - define the node struct
  - P2-2. point_search (search for a point that exists, and for a point that does not exist)
  - P2-3. range_search (find all points contained within a specified bounding rectangle)

Problem 2: Quad Tree Specification

- Assume the x, y values of all are integers between 1 and 20, and the original map is 21 x 21.

- Test build_tree function using your point set.
  - Create your own point set (more than twenty)

- Test the point_search function, by searching for the selected points and displaying the results.
  - At least three cases with your selected points.

- Test the range_search function with your selected rectangles and displaying the results.
  - At least three cases with your selected rectangles.

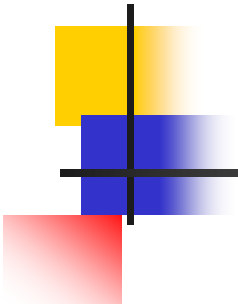
Submission Format

- Submit with a [Team-A].Zip file
  - /code/
    - Copy the entire visual studio project here.
    - comment the authors of each function in codes
  - Team-A.pdf (Documentation)
    - Team introduction (list all members & roles)
    - Contribution percentage (Kim: 25%, Park: 30% …)
    - Achievement table (self-evaluation)
    - Attach the Codes & Result screenshots for each problem

- Submission Due
  - By May 16th (Mon) 23:59 to CyberCampus

Achievement Table

Problem	Members Involved	Achievement
P1-1	name1, name2	100%
P1-2	name2	50%
P2-1	name3	0%
…	…	…



End of Lecture