CS042

Scalable Data Structure Study

Programmer's guide to the CS

Author:
Seungwoo Schin
Gihyo Jang
Seowon Oh

Typeset by:
Seungwoo Schin

PyDal study fork();

June 29, 2015

Contents

0. Introduction

I. Administrative Details

기본적인 데이터구조들을 여러 언어로 짜보고, 코딩한 결과를 서로 비교해 보는 걸 목적으로 합니다. 다른 스터디에서도 사용할 수 있도록 작성하였습니다. 기본적인 규칙들은 다음과 같습니다.

1. 전반적인 공지나 질문 등은 카톡방을 통해서 공유합니다.
2. 사용하는 언어는 자유이나, file io가 가능해야 합니다. 재귀, loop 등을 사용할 수 있으면 좋습니다.
3. 1주일에 한번씩 매주 토요일 10시부터 12시 반까지 만납니다.
4. 만날 때마다 전 주에 정해진 데이터구조에 대해서 한 명이 공부해서 아래 일들을을 하셔야 합니다.
   (a) 데이터구조에 대한 간략한 설명
   (b) 데이터구조를 txt 파일로 표현하는 방법에 대한 표준 설정
   (c) 해당 데이터구조에 대한 operation의 결과를 txt 파일로 나타내는 방법에 대한 표준 설정
   (d) 테스트용 파일과 솔루션 파일 작성
5. 만난 후 다음번에 만나기 전까지, 해당 스터디에서 커버한 부분에 대한 코드를 작성한 후, 테스트 결과와 같이 올려주세요.
6. 스터디장은 위 코드들과 테스트, 솔루션 파일들을 받아서 github에 업로드해야 하며, 스터디를 진행할 때마다 커버한 부분을 LaTeX으로 작성해서 올려야 합니다.
7. 스터디장은 LaTeX을 할 줄 알아야 합니다.

I.1 사용할 언어들

사용 언어는 자유이며, 아래의 언어는 사용 가능한 언어들의 hello world! 예제입니다. 아래에 없는 언어를 사용하고 싶으시면 스터디장에게 말해서 언어를 추가해 주세요.

1. C

```c
#include <stdio.h>

int main()
{
	printf("Hello World!\n");
}
```

2. Java

```java
public class Foo {
  public static void main(String args[]) {
    System.out.println("for test");
  }
}
```

3. Python

```
print "hello world!"
```

4. JavaScript

```
// Hello in Javascript

// display prompt box that ask for name and
// store result in a variable called who
var who = window.prompt("What is your name");

// display prompt box that ask for favorite color and
// store result in a variable called favcolor
var favcolor = window.prompt("What is your favorite color");

// write "Hello" followed by person' name to browser window
document.write("Hello " + who);

// Change background color to their favorite color
document.bgColor = favcolor;
```

II. Programming Basics

프로그래밍 언어와 독립적으로, 알고리즘을 평가하거나 디자인하는 방법에 대한 항목들입니다.

II.1 Timal/Spatial complexity

O(n)

II.2 Recursion

definition complexity Recursion example vs iteration stack overflow / tail-call recursion (+ continuation?)

II.3 Divide and Conquer

definition example

II.4 Dynamic Programming

introcd ../ example vs greedy

1. List

I. Structures to implement

I.1 Singly linked list

I.2 Doubly linked list

I.3 Circular linked list

II. Operations to implement

II.1 Sorting

II.2 Searching

II.3 Insertion

II.4 Deletion

2. Stack

I. Structures to implement

I.1 Simple stack

II. Operations to implement

II.1 Sorting

II.2 Searching

II.3 Insertion

II.4 Deletion

3. Queue

I. Structures to implement

I.1 Simple queue

I.2 Priority queue

I.3 Dequeue (double ended queue)

II. Operations to implement

II.1 Sorting

II.2 Searching

II.3 Insertion

II.4 Deletion

4. Advanced Structure

I. Structures to implement

I.1 Tree

Simple binary tree

Full binary tree

Perfect binary tree

Complete binary tree

Binary search tree

AVL tree

Red-black tree

Heap tree

B-tree

2-3-4 tree

B+ tree

I.2 Graph

Simple graph

Directed graph

Directed weighted graph

II. Operations to implement

II.1 Elementary algorithms

BFS

DFS

Topological sort

II.2 Minimum spanning tree

Prim

Kruskal

II.3 Shortest path algorithms

Bellman-Ford algorithms

Dijkstra's algorithm

Floyd-Warshall algorithm

Johnson's algorithm