

Structure of data and Algorithms

This class is about the basic algorithm in the field of computer science.

Impression

- I saw a lot of applications of recurrence formula that I learnt in the highschool. I did not know that some inequality evaluation problems in the university entrance exams are part of this field.
- I was surprised that really clever algorithms provide a platform for the library like `.Sort()`.
- However this class only taught us the basic algorithms so that we need to study extra in order to live in the computer science field.
- I mounted some algorithms using language C. The other classes gave us this opportunity, which helps us understand them.
- Tree structure are base of the almost all fast algorithms. I did not care so much about tree structure, but I realized its importance.

Memo

1. Algorithms and calculation amount
2. Basic structure of data
 - Array and list
 - difference between them
 - Stack
 - Queue
 - Tree
3. How to express set

- Priority queue and heap
 - insert: $O(\log n)$, delete: $O(\log n)$, delete: $O(n)$
- Binary search tree
 - $O(\log n)$ on average, in the worst case $O(n)$
- Balanced tree
 - $O(\log n)$
 - 2-3 tree
 - AVL tree
- Hash
 - chain
 - open address
- Set group
 - pointer
 - * find: $O(n)$ (can be $O(\log n)$ in an algorithm), merge: $O(1)$
 - ID
 - * find: $O(1)$, merge: $O(n)$

4. Sort

- Bubble sort
 - $O(n^2)$
- Quick sort
 - $O(n \log n)$ (worst case: $O(n^2)$)
- Merge sort
 - $O(n \log n)$
- Heap sort
 - when we need the minimums
 - swap and pushdown
 - $O(n + k \log n)$
- Bucket sort
- Radix sort

5. Directed graph

- Dijkstra
 - $O(n^2)$
- Dijkstra using heap

- $O((n + e)\log n)$, e is a number of edges
 - Floyd's algorithm
 - $O(n^3)$
 - Depth-first search
 - Breadth-first search
 - Strongly Connected Component
6. Undirected graph
- Prim's algorithm
 - $O(n^2)$
 - Kruskal's algorithm
 - $O(e\log e)$
 - Articulation point
7. Search system
- KMP
 - $O(n)$
 - BM
 - $O(n/m)$
 - Try tree
8. Algorithm design

日本語での感想

- 高校内容で習った漸化式の応用がたくさん出て来た。大学入試の数学などに出てくる不等式評価問題の一部はこのアルゴリズムが背景だったんだなと実感した。
- そもそも、ソートなどは`.sort()`などで気軽に行えるので、気にしていなかったが、中にはこういうアルゴリズムがあるのかと感動した。
- かなり基礎の基礎をやっているという印象。実験と競技プログラミングの授業とかぶるところがあってよかった。
- ほとんどのアルゴリズムが木に落とし込まれていた。木という構造はすごく便利なんだと感じた。