VE477 mid

Overall information

- Time: 2020/11/5, Thur. 18:20-20:00
- Location: F205
- Part A:
  - closed book
  - mentioned question + green slide questions
  - around 20 minutes
  - be quick
- 5 min break
- Part B:
  - open book
  - around 80 minutes
  - spend time wisely

Chapter 1

- Know what are they and their corresponding complexity of related operations
  - Dictionary
  - Binary search tree (balanced version, like AVL tree)
  - Hash table
  - Priority queue
- Others
  - MST
    - definition *1.31*
    - Prim/Kruskal(Union-find *1.38*) *1.39*
  - Sort and count

Chapter 2

Time complexity

- Worst case/best case/average case (like qsort)
- $\mathcal{O}, \Omega, \Theta$
- Ranking of different complexity *2.10*

Algorithm analysis

- Stable marriage problem *2.14,2.15*
- All remaining staff...

Computational complexity

- Different types of computational problems *2.35, 2.36*. Decision problem is the center of them.
- Formalizing complexity *2.46*
- Definition of $\mathcal{P}$ and $\mathcal{NP}$ *2.47, 2.48, 2.49*
- Complexity reduction
- Some traditional problems and their computational complexity (halting, SAT, 3-SAT, TQBF, Hilbert's tenth)
- Complexity diagram *2.65* (**important**)
  - Relation between $\mathcal{P}$, $\mathcal{NP}$, $\mathcal{NP}$-complete, $\mathcal{NP}$-hard
- Proof of $\mathcal{NP}$-complete (**important**) (ref to Ex.5&Ex.6 of hw4)
  - Step 1: prove the problem A is in $\mathcal{NP}$: prove that given a solution of a problem, we can verify that it is true in polynomial time (ref to Ex4. hw3)
  - Step 2: find another problem B, which is $\mathcal{NP}$-complete, and prove B can be reduced to A
    - Find a transformation f that maps instance of B to instance of A
    - Prove $\forall b \in L(B)$, we have $f(b) \in L(A)$
    - Prove $\forall f(b) \in L(A)$, we have $b \in L(B)$ OR $\forall b \notin L(B)$, we have $f(b) \notin L(A)$
    - Additional: **Here f can be a bijection function, but it is not necessary to be a bijection.**
      This means A is no easier than the hardest problem in $\mathcal{NP}$
  - Step 3: prove the reduction process f can be done in polynomial time
  - Conclude that A is in $\mathcal{NP}$-complete

Chapter 3

Graph related things

- Directed graph v.s. undirected graph: undirected graph is basically the same as a directed graph with bidirectional connections
- Dijkstra: cannot deal with negative edge
- Bellman-Ford: cannot deal with negative (undirected negative edge is also a negative cycle)

dp problems (important)

- Bellman-Ford
- Edit distance
- Partition problem (Ex.1 of hw5)
- Explore more by yourself

Chapter 4

Network flow

- Definition (*4.6*)
- Ford-fulkerson (*4.15*)
- Max-flow min-cut (*4.23*)
- Edmond-karp (a special case of FF, find path by BFS)

Application of network flow

- Bipartite matching (*4.43, 4.45*)
- wifi network problem (Ex.5 of hw5)

Homework

HW2

Ex1. 4. about a) and d)