Computational theory

1 Introduction 09/24

Today's class is about what we are going to learn in this class.

- We are going to learn the theoretical system about the difficulty of a specific problem
- Problems can be categorized into a class.
- Some problems are popular like TSP.
- Sorting problems are $O(nlogn)$ and $\Omega(nlogn)$
  - $f(n) = O(g(n))$ means $\exists n_0, \forall n \geq n_0, f(n) \geq g(n)$
  - $f(n) = \Omega(g(n))$ means $\exists n_0, \forall n \leq n_0, f(n) \geq g(n)$

2 What is P and NP problems 10/01

- What is "calculation"
  - Being able to calculate means there exists an algorithm that can be solved in a finite time with Turing machines
- Time and Space can be defined as follows
  - Let T be a number that represents how many times TM applys transition function
  - Let S be the number of cells that head went through
  - Time t(n) = max(T(n)), Space s(n) = max(S(n))
  - if t(n) is polynomial, that problem is called P.
- Nondeterministic turing machine
  - there exists more than one transition function
  - this express NP

3 Complexity class 10/08

- Proof of "$P \subseteq NP \subseteq PSPACE \subseteq EXPTIME$"
- complexity class
  - we define some classes in NP in order to discuss NP problems
  - Two important classes, SAT and HC.

- polynomial-time reduction, used to define order in NP classes
- $SAT \leq_M HC$ and $HC \leq_M SAT$

4 NP completeness 10/15

- SAT
  - SAT is a class of problems whose inputs are CNF propositional formula and outputs are its satisfiability.
  - it is included in the NP-complete
- HC $\leq_m$ TSP
  - HC is considered to be a TSP by giving a cost to each branch.
  - if a branch is existed in HC give 1 to it and if it doesn't give 2 to it.
  - G has HC $\Leftrightarrow$ K_n has HC whose length is n.
- Definition of NP-complete Π
  - $\Pi \in NP$
  - $\forall \Pi' \in NP$, $\Pi' \leq_m \Pi$
- Cook-Levin theorem
  - SAT $\in$ NP-complete

5 NP completeness 2 10/29

- Encoding and input size
- Polynomial-time reduction
- prove 3SAT $\in$ NP-complete
  - First, prove 3SAT $\in$ NP
  - Second, prove $\Pi' \leq_m \Pi$
- Prove VC $\in$ NP complete

6 NP completeness 3 11/05

- Define NTM again in the more genuinely using a guessing module
- Three ways to prove the NP-completeness
  - Restriction
    - Problem classes which has NP complete problems as the partial problems are NP complete
    - Subgraph Isomorphism
  - Local Replacement
    - Redunction from SAT to 3SAT
  - Component Design

7 Various problems 11/12

- There are some problems that seem to be similar but quite different in terms of complexity
  - shortest path problems vs longest path problems
  - edge cover vs vertex cover
  - minimum cut vs maxium cut
- partition
  - which is important to consider the integral numbers problems

8 Partition 11/19

- partition problems
  - instance : $a_1, a_2, ..., a_n$
  - question : $\Sigma_{i \in S} a_i = \Sigma_{j \in [n]-S} a_j$
- Dynamic programming is on of the ways to solve the partition problems
- definition of pseudo polynomial time algorithms
  - polunomial time about the input size and max a_i
- Strong completeness
  - Even when max a_i is n's polynomial, still NP complete

9 Approach to NP-completeness 11/26

we consider the approach to the NP-completeness class. To solve it exactly seems to be difficult thus we try to solve it approximately.

- Bin packing
  - first fit FF
  - FFD, make laguages in order of their sizes
  - $\forall I.FF(I) \leq \frac{17}{10} OPT(I) + 2$
  - $\exists I.FF(I) \geq \frac{17}{10}(OPT(I) - 1)$
  - $\forall I.FFD(I) \leq \frac{11}{9} OPT(I) + 4$
  - $\exists I.FFD(I) \geq \frac{11}{9} OPT(I)$

10 Metaheuristics and TSP 12/03

- some heuristics
  - local search
  - simulated anealing

  - genetic algorithms
  - tabu search
  - quantum anealing
- algorithms to solve metric TSP
  - find minimum weight spanning tree T
  - double the edges of T
  - find eulerian circuit
  - create hamilton circuit
- algorithms to solve metric minimum weight matching

11 PTAS and knapsack 12/10

- definition of Polytime approximation scheme (PTAS)
  - $\forall \epsilon$, there is a polytime algorithm whose approximation ratio is $1 + \epsilon$
  - There is no ptas algorithm in general TSP problems, but there is in Euclidean TSP
- knapsack problem
  - Mitigation problem
  - Greedy algorithm
  - Patal enumeration

12 PTAS and FPTAS 12/17

- FPTAS
  - $\forall \epsilon$, there is a polytime algorithm whose approximation ratio is $1 + \epsilon$, with respect of input size n and $\frac{1}{\epsilon}$
- Scaling
- if there is FPTAS algorithm then there is pseudo polytime algorithm