Advanced Algorithm

Jialin Zhang
zhangjialin@ict.ac.cn

Institute of Computing Technology, Chinese Academy of Sciences

March 14th, 2019

Course Information

- Instructor: Jialin Zhang, zhangjialin at ict dot ac dot cn
- TA: Rui Zhang, zhangrui2016 at ict dot ac dot cn
- URL: `http://z14120902.github.io/alg.html`
- Office hours: Arrange by email
- Lecture: Thursday, 2:00; 4th floor, ICT
- Grades: Homework 40%, Final Exam (closed-book) 40%, Presentation 20%

Course Contents

Randomized Algorithm complexity classes, inequalities, balls and bins (Inequalities), probabilistic method ...

Approxidmation Algorithm combinatory algorithm, LP-based approximation algorithm, hardness of approximation...

Quantum computing quantum algorithm, Grover's algorithm, Shor's algorithm...

Reference Book:

- *Randomized Algorithm* by R.Motwoni and P.Raghavan
- *Approximation Algorithm* by V.Vazirani
- *Quantum Computation and Quantum Information* by Michael A. Nielsen, Isaac L. Chuang

Lecture 1: Introduction of Randomized Algorithm

Outline

1. Big O notation
2. Quick Sort
3. Min Cut

Big O notation

Big O notation

1. $f(n) = O(g(n))$: $\leq$
2. $f(n) = o(g(n))$: $<$
3. $f(n) = \Omega(g(n))$: $\geq$
4. $f(n) = \omega(g(n))$: $>$
5. $f(n) = \Theta(g(n))$: $=$

Big O notation

Big O notation

1. $f(n) = O(g(n))$: $\exists$constant $c, \overline{\lim}_{n\to\infty} \frac{f(n)}{g(n)} \leq c$
2. $f(n) = o(g(n))$: $\overline{\lim}_{n\to\infty} \frac{f(n)}{g(n)} = 0$
3. $f(n) = \Omega(g(n))$: $g(n) = O(f(n))$
4. $f(n) = \omega(g(n))$: $g(n) = o(f(n))$
5. $f(n) = \Theta(g(n))$: $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$

Sorting Problem

Ref: Randomized Algorithm - Chapter 1, Thm 1.1.

Sorting Problem: Given a set S of n numbers, sort them into ascending order.

- First trial
  1. Find median y of set S; (cn)
  2. Partition $S \setminus \{y\}$ into two sets S_1 and S_2; (n)
  3. Recursively sort S_1 and S_2.
  4. Time complexity: $T(n) \leq 2T(n/2) + (c+1)n$.
  5. But: median is complicated to find.
- Second trial
  1. Find y such that two sets S_1 and S_2 are approximately same size;
  2. For example $T(n) \leq T(n/4) + T(3n/4) + (c+1)n$.

Quick Sort

Ref: Randomized Algorithm - Chapter 1, Thm 1.1.

Sorting Problem: Given a set S of n numbers, sort them into ascending order.

- Quick Sort
  1. Find random y of set S; $(O(1))$
  2. Partition $S \setminus \{y\}$ into two sets S_1 and S_2; (n)
  3. Recursively sort S_1 and S_2.
  4. Time complexity: ???
- $O(n \log n)$ does not depend on input. It holds for every input.

Min Cut

- Ref: Randomized Algorithm - Chapter $1.1, 10.2$.
- **Min-Cut problem**: Given a graph $G = (V, E)$ which is a connected, un-directed multi-graph, find a cut with minimum cardinality.
- standard method: using max flow algorithm
  - Ford-Fulkerson algorithm: $O(|E||f^*|)$;
  - Edmonds-Karp algorithm: $O(|E|^2|V|)$;
  - Dinic algorithm: $O(|E||V|^2)$;
  - Push-relabel algorithm: $O(|E||V|^2)$;
  - etc. . .

A Randomized Algorithm for Min-Cut Problem

Contrast Algorithm:

1. Pick an edge uniformly at random;
2. Merge the endpoints of this edge;
3. Remove self-loops;
4. Repeat steps 1-3 until there are only two vertices remain.
5. The remaining edges form a candidate cut.

Min Cut

- What is the successful probability?
  - $\Omega(\frac{1}{n^2})$
  - Repeat the contrast algorithm $\Theta(n^2)$ times, successful probability : constant
- Time complexity?
  - Time complexity is $O(n^2)$.
  - How to uniformly randomly pick an edge?

Homework

1. In QuickSort algorithm, based on the recursion, prove the expected running time is $O(n \log n)$.
2. Randomized Algorithm - Exercise 1.2, Page 9.