Mathematics for Computer Science (30470023) Spring 2020

Lecture 6 — Mar 23, 2020

Instructor: Prof. Andrew C. Yao *Scribes: Chengming Shi, Yizhi Huang*

1 Overview

In this lecter we introduced *the Borodin-Hopcroft lower bound* of the bit-fixing routing algorithm, which implies the limitation of the deterministic oblivious routing and the necessity of randomization. Then we introduce the generating function, one of the most useful inventions in Discrete Math.

2 Main Lemma

Lemma 2.1 (Main Lemma). In the randomized routing algorithm on a n-cube network, which is introduced last week, we fix i and $\rho(i)$ and pick $\rho(j)$ for $j \neq i$ randomly and independently, then

$$\Pr\{\text{Delay of message } M_i > 6n\} \leq 2^{-4n}. \qquad \diamondsuit$$

To prove the main lemma, we introduce the second lemma.

Lemma 2.2 (Main Lemma 2). Take any path $P = e_1, \dots, e_l$ by bit-fixing and pick $\rho(j)$ for all $j \in \{1, \dots, N\}$, then

$$\Pr\{|S| > 6n\} \leq 2^{-4n}$$

where $S = \{j \mid Path_j \cap P \neq \emptyset\}$. $\diamondsuit$

Exercise 3 in HW5 shows that the second lemma guarantees the main lemma. Thus we just need to prove the second one.

Proof. Set random variables

$$X_j = \begin{cases} 1 & \text{If } Path_j \cap P \neq \emptyset \\ 0 & \text{Otherwise} \end{cases}$$

and then $|S| = \sum_j X_j$. By Chernoff Bound, our mission is to show $E(|S|) = O(n)$. Since P can be represented by $e_1, e_2, \dots, e_n$, we set another random variables Y_{e_k} denoting the number of $Path$ containing the edge e_k. It's clear to see

$$E(|S|) \leq \sum_{k=1}^{l} E(Y_{e_k}).$$

where $E(Y_{e_k})$ is independent of k because all $\rho(j)$ are picked independently. We can rewrite $E(Y_{e_k}) = E(Y_e)$ and

$$E(|S|) \leq l \cdot E(Y_e).$$

To calculate $E(Y_e)$, we count the number of pairs (j, e) where $Path_j$ contains e. The expectation of number of pairs can be calculated either by edge or by node. Counting by edge, it is

$$\begin{aligned} E(\text{number of pairs}) &= \sum_{\text{edge } e} E(Y_e) \\ &= E(Y_e) \cdot (\text{number of edge}) \\ &= E(Y_e) \cdot Nn. \end{aligned}$$

Note that each undirected edge should be counted twice because each edge can provide simultaneously two passways, different in directions. Counting by node, the expectation is

$$E(\text{number of pairs}) = \sum_{\text{node } j} E(\text{length of } Path_j) = N \cdot \frac{n}{2}.$$

The length of each path is the Hamming distance between the source and the destination, and thus its expectation is $n/2$. Now we can get

$$E(Y_e) = \frac{1}{2}$$

and

$$E(|S|) \leq \frac{l}{2} \leq \frac{n}{2}.$$

By Chernoff Bound,

$$\Pr\{|S| > t = (1+\delta)\mu\} \leq \frac{1}{1+\delta}\left(\frac{e}{1+\delta}\right)^{t-\mu} \leq \left(\frac{e}{1+\delta}\right)^{t}.$$

Let $t = 6n$. Since $\mu = E(|S|) \leq n/2$, $\delta \geq 5.5$,

$$\Pr\{|S| > t = (1+\delta)\mu\} \leq \left(\frac{e}{1+\delta}\right)^{6n} \leq 2^{-4n}.$$

□

As mentioned before, the second lemma guarantees the main lemma 2.1.

3 The Borodin-Hopcroft Lower Bound

A *permutation routing problem* is a problem in which every node is the source of one source-destination pair and the destination of one pair. Thus, the routing problem can be represented by a permutation σ on the set of nodes.

Theorem 3.1 (The Borodin-Hopcroft Lower Bound). Any deterministic oblivious routing on a N-size graph where the maximal degree of edges is d will have $\sqrt{N/d}$ congestion. ♢

To be specific, for every deterministic oblivious routing strategy, there is a permutation σ that causes an overlap of at least $\sqrt{N/d}$ paths at some node.

Proof. We just provide the sketch of the proof.

First, convert the undirected graph into a directed graph where each edge is replaced by two directed edges of opposite directions. Then, for any oblivious algorithm A, find a node i s.t. it's "congested" by some permutation σ, which means there exist $\sqrt{N/d}$ nodes k s.t. $Path_{k\to\sigma(k)}$ all go through i.□

To do such things, we introduce a top-level proof idea.

Definition 3.2. Let t be a node, and a node i is a *gateway* to t if there exist $\sqrt{N/d}$ $Path_{k \to t}$ containing i. ♢

By the definition of *gateways*, we provide two lemmas to help to prove the theorem.

Lemma 3.3. For each t, there are at least $\frac{1}{2}\sqrt{N/d}$ gateways to it. ♢

Lemma 3.4. There exist a node i that is a gateway to $\frac{1}{2}\sqrt{N/d}$ destinations. ♢

Then we can prove the theorem by contructing a set of $\frac{1}{2}\sqrt{N/d}$ $Path_{k \to t_k}$ through i with distinct pairs.

Theorem 3.1 is of particular interest for the following reasons:

- It gives a reason for adding randomisation into our consideration to search for algorithms, as a deterministic algorithm may not work efficiently, as proved above;
- It gives a lower bound for the efficiency of a class of algorithms, which in general is hard to derive.

4 Generating Function

4.1 Definitions and basic properties

Definition 4.1. Let $A = \{a_i\}_{i \in \mathbb{N}}$ be a sequence of real numbers, and suppose the power series $\sum_{i=0}^{\infty} a_i x^i$ uniformly converges in a neighbourhood of 0. Define the *generating function* of A to be

$$f_A(x) = \sum_{i=0}^{\infty} a_i x^i.$$

♢

Remark 4.2. Here we define the generating function as a function in common sense. Alternatively, one may define a generating function of A to be a *formal sum*

$$f_A(X) = \sum_{i=0}^{\infty} a_i X^i$$

where X is an indefinite not in $\mathbb{R}$. Such formal sums are called *formal power series*. From this perception, the generating function can be defined without the requirement of uniform convergence, and the properties of addition and multiplication given below become definitions. But we usually consider only those generating functions that uniformly converges in a neighbourhood of 0, so the two definitions given here are in effect equivalent.

Proposition 4.3 (Addition and multiplication).

$$\sum_{i=0}^{\infty} a_i x^i + \sum_{i=0}^{\infty} b_i x^i = \sum_{i=0}^{\infty} (a_i + b_i) x^i$$

$$\left(\sum_{i=0}^{\infty} a_i x^i\right)\left(\sum_{i=0}^{\infty} b_i x^i\right) = \sum_{i=0}^{\infty}\left(\sum_{j=0}^{i} a_j b_{i-j}\right) x^i$$

$\{\sum_{j=0}^{i} a_j b_{i-j}\}_{i\in\mathbb{N}}$ here is called the *convolution* of $\{a_i\}_{i\in\mathbb{N}}$ and $\{b_i\}_{i\in\mathbb{N}}$. ♢

Proposition 4.4 (Derivative and integral). Let the generating function of $A = \{a_i\}_{i\in\mathbb{N}}$ be f_A, then

$$f_A'(x) = \sum_{i=0}^{\infty} (i+1) a_{i+1} x^i$$

$$\int f_A(x)\,\mathrm{d}x = \sum_{i=1}^{\infty} \frac{a_{i-1}}{i} x^i + C$$

♢

These properties can easily seen to be true. For mathematically rigorous proofs, refer to any book on mathematical analysis.

4.2 Common generating functions

To obtain the corresponding sequence of a generating function, we need to obtain its power series expansion at 0. Without mathematical rigour, it's just the Taylor expansion at 0.

Proposition 4.5 (Binomial theorem).

$$(1+x)^n = \sum_{i=0}^{n} \binom{n}{i} x^i$$

and more generally

$$(a + bx^m)^n = \sum_{i=0}^{n} \binom{n}{i} a^{n-i} b^i x^{im}$$

♢

Proposition 4.6.

$$\frac{1}{1-x} = \sum_{i=0}^{\infty} x^i$$

and much more generally

$$\frac{1}{(1-bx^m)^n} = \sum_{i=0}^{\infty} \binom{i+n-1}{n-1} b^i x^{im}$$

♢

Proof. For the latter equation, when $n = 1$ it's not difficult to prove. Then, use mathematical induction on n, and note that

$$\sum_{j=0}^{i} \binom{i+n-2}{n-2} = \binom{i+n-1}{n-1}. \qquad \square$$

We can now calculate the corresponding sequence of any rational generating function by partial fraction decomposition[1]. This technique would be useful when we use generating functions to solve linear recurrence relations, as will be discussed later.

4.3 Applications

Generating functions links two difference objects of mathematics — sequences and functions, thus providing each the powerful tools used on the other side, resulting in fruitful applications, of which we name some below.

Example 4.7 (Variance of random variables). Let $X_i, 1 \le i \le n$ be independent random variables, each with probability b to be 1 and otherwise being 0. Let $X = \sum_{i=1}^{n} X_i$. Let $a_i = \Pr[X = i]$ and $A(x) = \sum_{i=0}^{n} a_i x^i$. Then A has the following property:

1. $A(x) = \sum_{i=0}^{n} \binom{n}{i} (1-b)^{n-i} b^i x^i = (1-b+bx)^n$, so $A'(x) = nb(1-b+bx)^{n-1}$ and $A''(x) = n(n-1)b^2(1-b+bx)^{n-2}$;
2. $A(1) = \sum_{i=0}^{n} a_i = 1$;
3. $A'(1) = \sum_{i=0}^{n} i a_i = \mathbb{E}(X)$;
4. $\mathrm{Var}(X) = \mathbb{E}(X^2) - (\mathbb{E}(X))^2 = \sum_{i=0}^{n} i^2 a_i - (\sum_{i=0}^{n} i a_i)^2 = A''(1) + A'(1) - (A'(1))^2$.

Therefore, $\mathrm{Var}(X) = n(n-1)b^2 + nb - (nb)^2 = nb(1-b)$.

Example 4.8 (Tile covering). Consider a $2 \times n$ rectangular tile. We want to use n dominoes to cover the all the tiles without intersection. Denote by a_n the number of ways to do that. Clearly,

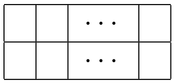

Figure 1: A $2 \times n$ rectangle

$a_0 = 1$ and $a_1 = 1$. For a_n, consider how the dominoes cover the rightmost tiles. It may be 2 horizontal dominoes or a vertical domino. In the former case, a $2 \times (n-2)$ rectangle remains, and there are a_{n-2} ways to cover it; whereas in the latter case, a $2 \times (n-1)$ rectangle remains, with a_{n-1} ways to cover it. We thus obtain $a_n = a_{n-1} + a_{n-2}$, so $\{a_i\}_{i \in \mathbb{N}}$ is just the Fibonacci sequence with an offset 1.

[1] Here it may involve complex numbers, which usually do not matter however, since we'll deal only with complex-*valued* functions, not those with complex *variables*.

Here we give an altenative way to calculate $\{a_n\}$. Let its generating function to be f. Note that

$$\sum_{i=2}^{\infty} a_i x^i = \sum_{i=2}^{\infty} a_{i-1} x^i + \sum_{i=2}^{\infty} a_{i-2} x^i,$$

we obtain $f(x) - a_0 - a_1 x = x(f(x) - a_0) + x^2 f(x)$, *i.e.*,

$$\begin{aligned} f(x) = \frac{1}{1-x-x^2} &= \frac{\sqrt{5}+1}{2\sqrt{5}(1-\frac{\sqrt{5}+1}{2}x)} + \frac{\sqrt{5}-1}{2\sqrt{5}(1-\frac{-\sqrt{5}+1}{2}x)} \\ &= \sum_{i=0}^{\infty} \left(\frac{1}{2\sqrt{5}} \left(\frac{\sqrt{5}+1}{2} \right)^{i+1} - \frac{1}{2\sqrt{5}} \left(\frac{-\sqrt{5}+1}{2} \right)^{i+1} \right) x^i, \end{aligned}$$

which gives the value of a_i.

The example above can be generalised as below.

Example 4.9 (Linear recurrence relation). Let $b_0, b_1, \dots, b_{k-1} \in \mathbb{R}$. Suppose $\{a_i\}_{i\in\mathbb{N}}$ satisfies $a_{i+k} = \sum_{j=0}^{i-1} b_j a_{i+j}$ for any $i \in \mathbb{N}$. Let f be the generating function of $\{a_i\}_{i\in\mathbb{N}}$, then

$$f(x) - \sum_{i=0}^{k-1} a_i x^i = \sum_{j=0}^{k-1} b_j x^{k-j} \left(f(x) - \sum_{i=0}^{j-1} a_i x^i \right),$$

whence it follows that $f(x)$ is a rational function, hence a sum of partial fractions, whose corresponding sequences we have already obtained in Proposition 4.6.

The generating function is also a powerful tool in proving combinatorial identities. Here we give an example in the homework of week 2.

Example 4.10 (LPV Problem 3.8.8). Prove the following identity:

$$\sum_{k=0}^{n} \binom{n}{k}\binom{k}{m} = \binom{n}{m} 2^{n-m}.$$

Proof. Note that

$$\sum_{m=0}^{\infty}\sum_{k=0}^{n} \binom{n}{k}\binom{k}{m} x^m = \sum_{k=0}^{n} \left(\binom{n}{k} \sum_{m=0}^{\infty} \binom{k}{m} x^m \right) = \sum_{k=0}^{n} \binom{n}{k} (x+1)^k = (x+2)^n = \sum_{m=0}^{n} \binom{n}{m} 2^{n-m} x^m.$$

Therefore, by comparing the coefficient of x^m on both sides, we obtain the identity. □

Another proof using generating function:

Proof. Note that

$$\begin{aligned} \sum_{n=0}^{\infty}\sum_{k=0}^{n} \binom{n}{k}\binom{k}{m} x^n = \sum_{k=0}^{\infty} \binom{k}{m} x^k \sum_{n=k}^{\infty} \binom{n}{k} x^{n-k} &= \sum_{k=0}^{\infty} \binom{k}{m} \frac{x^k}{(1-x)^{k+1}} \\ &= \frac{x^m}{(1-x)^{m+1}} \sum_{k=0}^{\infty} \binom{k}{m} \left(\frac{x}{1-x} \right)^{k-m} = \frac{x^m}{(1-x)^{m+1}(1-x/(1-x))^{m+1}} \\ &= \frac{x^m}{(1-2x)^{m+1}} = \sum_{n=0}^{\infty} \binom{n}{m} 2^{n-m} x^n. \end{aligned}$$

Therefore, by comparing the coefficient of x^n on both sides, we obtain the identity. □