Thursday, February 8, 2018

Application of Prufer Code: Random Tree Generation, Cayley's Formula and Building Tree from Degree Count

On the last post (Prufer Code: Linear Representation of a Labeled Tree), we discussed how to convert a labeled tree into Prufer Code and vice versa. On this post, we will look into some of its applications in problem-solving.

Generating Random Tree

This one is the simplest. Though problem-solving may not require generating random trees, problem setting might. In order to generate a random tree with N nodes, all we need to do is generate a Prufer Code of length $N-2$ with each element being in the range of 1 to N. Next, we just convert the Prufer Code into a tree.

The randomness of the tree depends on the randomness of the sequence generated. Using `rand()` function is not good since it's not truly random. How we can generate a random sequence requires a blog post of its own. Maybe I will write a post on it in near future.

Cayley's Formula

Given N labeled nodes, how many spanning trees can we build?

For example, how many different spanning trees can we build with 3 nodes?

Follow by Email

Email address... Submit

Labels

Analysis Arithmetic Function Backtrack Big Int Binary Bitwise Combinatorics Complexity Contest CPPS D&C Divisors Factorial Factorization GCD Graph Language LCM Logarithm Math Modular Arithmetic Number Theory Optimization Primality Test Prime Proof Repeated Squaring Sequence Sieve SPOJ Theorem Tree UVa

Blog Archive

▼ 2018 (2)
▼ Feb (1)
Application of Prufer Code: Random Tree Generation...
► Jan (1)
► 2017 (2)
► 2015 (35)

According to Cayley's Formula, the number of spanning trees we can build with N labeled node is N^{N-2}.

Cayley's Formula can be proved by Prufer Code easily. Since there are N nodes, what will be the length of the Prufer Code of the spanning trees? Ans: $N-2$. Next, at each position of the Prufer Code, how many different values can we place? Ans: N. Therefore, we have N^{N-2} combinations possible for Prufer Code.

Building Tree from Degree Count

Given N nodes and for each node its degree count, i.e, number of other nodes that are its neighbor, how many different spanning trees can we build while maintaining the degree count of each node?

For example, let $N=5$ and the degree count be $[3,2,1,1,1]$. How many spanning trees respect these degree constraints? Ans: 3

So how do we find the answer using Prufer Code? We use the following two information:

- A labled tree with N nodes has a Prufer Code of length $N-2$.
- If a node has degree count d, then it appears exactly $d-1$ times in the Prufer Code.

So for each node, we know how many times it appears in Prufer Code from its degree count. If the degree count for i_{th} node is d_i, then it appears $d_i - 1$ times. We also know the total number of openings we have in our Prufer Code ($N-2$ openings). So the number of possible spanning tree while maintaining the degree constraint is:

$$\binom{N-2}{d_1-1}\binom{N-2-(d_1-1)}{d_2-1}\dots\binom{N-2-(d_1-1)-\dots-(d_{N-1}-1)}{d_N-1} = \binom{N-2}{d_1-1, d_2-1, \dots, d_N-1}$$

For example, when $N=5$ and degree count is $[3,2,1,1,1]$, the length of Prufer Code will be $5-2=3$ and the nodes will appear $[3-1, 2-1, 1-1, 1-1, 1-1] = [2,1,0,0,0]$ times. So answer will be $\binom{3}{2,1,0,0,0} = \binom{3}{2}\binom{1}{1} = 3$.

Simple right? See if you can solve the following problem now: UVa 12956 - Curious Guardians. It's going to require DP along with what we have learned so far.

Conclusion

I know about one more application of Prufer Code, but I did not write about it here because its proof is a bit tricky. So I will probably write a separate post for that. If you are feeling curious about it then here is a spoiler: UVa 11719 - Gridland Airports.

If you have enjoyed learning Prufer Code, then don't forget to share it.

Resources

1. Wiki - Caley's Formula
2. forthright48 - Prufer Code: Linear Representation of a Labeled Tree

Related Problems

1. UVa 12956 - Curious Guardians

Posted by Mohammad Samiul Islam

Labels: Combinatorics, Graph, Proof, Theorem, Tree

No comments:

Post a Comment

Leave comments for Queries, Bugs and Hugs.

Enter your comment...

Comment as: Google Accoun

Publish Preview

Home Older Post

Subscribe to: Post Comments (Atom)

Popular Posts

SPOJ LCMSUM - LCM Sum
Problem Problem Link - SPOJ LCMSUM Given n, calculate the sum $LCM(1,n) + LCM(2,n) + \ldots + LCM(n,n)$, where $LCM(i,n)$ denotes the ...

Euclidean Algorithm - Greatest Common Divisor
Problem Given two number A and B, find the greatest number that divides both A and B. What we are trying to find here is the Greatest Comm...

Extended Euclidean Algorithm
Extended Euclidean Algorithm is an extension of Euclidean Algorithm which finds two things for integer a and b: It finds the value of...

Chinese Remainder Theorem Part 1 - Coprime Moduli
Second part of the series can be found on: Chinese Remainder Theorem Part 2 - Non Coprime Moduli Wow. It has been two years since I pub...

Prufer Code: Linear Representation of a Labeled Tree
I guess this is going to be my first post (apart from the contest analysis') which is not about Number Theory! It's not about graph ...

Segmented Sieve of Eratosthenes
Problem Given two integers A and B, find number of primes inside the range of A and B inclusive. Here, $1 \leq A \leq B \leq 10^{...

Sieve of Eratosthenes - Generating Primes
Problem Given an integer N, generate all primes less than or equal to N. Sieve of Eratosthenes - Explanation Sieve of Eratosthenes ...

Number of Digits of Factorial
Problem Given an integer N, find number of digits in $N!$. For example, for $N = 3$, number of digits in $N! = 3! = 3\times 2\times 1...

Euler Totient or Phi Function

I have been meaning to write a post on Euler Phi for a while now, but I have been struggling with its proof. I heard it required Chinese Rem...

Chinese Remainder Theorem Part 2 - Non Coprime Moduli
As promised on the last post, today we are going to discuss the "Strong Form" of Chinese Remainder Theorem, i.e, what do we do whe...