

SCB-PA Inter School and College Programming Contest 2019

March 6, 2020

A. Tom Is a Good Mentor

Setter: হাসিনুর রহমান

বলতে হবে $[1, M]$ রেঞ্জের ভেতর কতগুলো X আছে যেন $X \times Z \neq Y \times N$ যেখানে $1 \leq Y < X$ এবং $1 \leq Z < N$ । এখানে X এবং Y অবশ্যই পূর্ণসংখ্যা।

Solution

আমরা বলতে পারি $XN = XN$ । এখান থেকে আমরা লিখতে পারি

$$X * (N/\gcd(X, N)) = (X/\gcd(X, N)) * N$$

ধরি $Z = N/\gcd(X, N)$ এবং $Y = X/\gcd(X, N)$ । যদি $\gcd(X, N) > 1$ হয়, তাহলে $Z = N/\gcd(X, N) < N$ এবং $Y = X/\gcd(X, N) < X$ ।

কাজেই আমরা বলতে পারি $\gcd(X, N) > 1$ হলেই কেবল $X \times Z \neq Y \times N$ শর্তটি মিথ্যা হয়। তাই শর্তটি সত্য হওয়ার জন্য $\gcd(X, N) = 1$ হতে হবে। তার মানে আমরা এমন সব X এর সংখ্যা গণনা করব যারা N এর সাথে সহমৌলিক (Co-prime)

Subtask 1

প্রথম সাবটাস্কের জন্য যেখানে $M = N$ সেখানে উত্তর হবে $\phi(N)$

Subtask 2

দ্বিতীয় সাবটাস্কের জন্য যেখানে $M = 0 \pmod{N}$ সেখানে উত্তর হবে $(M/N) \times \text{phi}(N)$

Subtask 3

তৃতীয় সাবটাস্কের জন্য যেখানে M, N দ্বারা বিভাজ্য নাও হতে পারে সেখানে N এর প্রাইম ফ্যাক্টরগুলো বের করতে হবে। এরপর ইনক্লুশন এক্সক্লুশন করে বের করতে হবে M এর ভেতরে এমন কতগুলো সংখ্যা আছে যেখানে N এর প্রাইম ফ্যাক্টরের কোনটিই নেই। যেসব সংখ্যায় N এর কোন প্রাইম ফ্যাক্টর উপস্থিত নেই সেগুলোকে আমরা N এর সাথে কোপ্রাইম বলতে পারি।

B. Konami Code

Setter: মাহমুদ রিদোয়ান

কনামি কোড প্রবলেমটি মূলত একটি স্ট্রিং ম্যাচিং প্রবলেম। তোমাকে একটি টেক্সট দেওয়া থাকবে এবং একটি প্যাটার্ন দেওয়া থাকবে। তোমাকে বলতে হবে, টেক্সটটিতে প্যাটার্নটি কতবার পাওয়া যায়।

Solution

সমস্যাটির লিমিট ছোট হবার কারণে ব্রুট ফোর্স পদ্ধতিতেই প্রবলেমটা সমাধান করা যাবে। টেস্টকেসটির প্রতিটা অবস্থান থেকে শুরু করে দেখতে হবে যে প্যাটার্নটি হুবহু মিলে যায় কি না। যদি মিলে যায় তাহলে যোগফল এক বাড়াতে হবে।

C. Palindrome and Queries

Setter: মোঃ মশিউর রহমান

N টা স্ট্রিং এর একটি অ্যারে A এবং Q টা কুয়েরি দেয়া হবে। প্রতি কুয়েরিতে সবচেয়ে বড় প্যালিনড্রোমিক সাবস্ট্রিং এর লেংথ বের করতে হবে যেটা কমপক্ষে L টা স্ট্রিং এবং সর্বোচ্চ R টা স্ট্রিং এর মধ্যে আছে। যদি সেরকম কোনো প্যালিনড্রোমিক সাবস্ট্রিং না থাকে, তাহলে 0 প্রিন্ট করতে হবে।

Subtask 1

ব্রুটফোর্স করে প্যালিনড্রোমিক সাবস্ট্রিংগুলো বের করে প্রতি স্ট্রিং এর ইউনিক প্যালিনড্রোম গুলার কাউন্ট বাড়িয়ে প্রতি কুয়েরিতে L এবং R রেঞ্জের সাবস্ট্রিং গুলোর মধ্যে লেংথের ম্যাক্সিমাম টা প্রিন্ট করতে হবে। স্ট্রিং কাউন্টের জন্য স্ট্রিং কি হিসেবে রেখে ম্যাপ ব্যবহার করা যেতে পারে।

Subtask 2

ডিপি করে প্যালিনড্রোমিক সাবস্ট্রিংগুলো বের করে প্রতি স্ট্রিং এর ইউনিক প্যালিনড্রোম গুলার কাউন্ট বাড়িয়ে প্রতি কুয়েরিতে সেগমেন্ট ট্রি দিয়ে L এবং R রেঞ্জের সাবস্ট্রিং গুলোর মধ্যে লেংথের ম্যাক্সিমাম টা প্রিন্ট করতে হবে। কাউন্টের জন্য হ্যাশ+আন অর্ডার ম্যাপ ব্যবহার করতে হবে।

Subtask 3

সব স্ট্রিংগুলো মার্জ করে প্যালিনড্রোমিক ট্রি দিয়ে কোন প্যালিনড্রোম কোথায় শুরু হয়েছে তার ভিত্তিতে কোন প্যালিনড্রোম কতটা স্ট্রিং এ আছে সেটার কাউন্ট বের করতে হবে। এর পরে কাউন্টকে ইন্ডেক্স ধরে লেংথ এর সেগমেন্ট ট্রি বিল্ড করে প্রতি কুয়েরিতে রেঞ্জের ম্যাক্সিমাম বের করতে হবে।

D. Crypto-Number

Setter: রিসাল শাহরিয়ার শেফিন

শুরুতে একটা এয়ারে দেওয়া থাকবে। প্রবলেমে ক্রিপ্টো নাম্বার নামক একধরনের নাম্বারের কথা বলা হয়েছে। কুয়েরীতে ক্রিপ্টো নাম্বার দেওয়া থাকবে। বলতে হবে এই ক্রিপ্টো নাম্বারের কতগুলো ডিভাইসর এয়ারেতে আছে।

Subtask 1

প্রতি কুয়েরীতে এ্যারে ট্রাভার্স করে ত্রিপ্টো নাম্বারের ডিভাইসর কোনগুলো সেগুলো চেক করে কাউন্ট করা। কম্প্লেক্সিটিঃ $O(QN)$.

Subtask 2

শুরুতে আনঅর্ডার্ড ম্যাপ বা এমন কিছু দিয়ে এ্যারের কোন নাম্বার কতবার আছে, সেটার কাউন্ট রাখতে হবে। এরপর ক্রিপ্টো নাম্বারের ডেফিনিশন লক্ষ্য করলে দেখা যায়, কোনো ক্রিপ্টো নাম্বারের যেকোনো প্রাইম ডিভাইসর p এর জন্য, ক্রিপ্টো নাম্বারটা অন্ততপক্ষে p^2 দিয়ে ডিভাইজিবল। তার মানে ক্রিপ্টো নাম্বারের বর্গমূলের ভিতরই সব প্রাইম ডিভাইজর রয়েছে। এই প্রোপার্টি ব্যবহার করে প্রতি কুয়েরীতে ক্রিপ্টো নাম্বারের বর্গমূল পর্যন্ত নাম্বার নিয়ে প্রাইম ফ্যাক্টোরাইজ করে ফেলা সম্ভব। এরপর প্রাইম ফ্যাক্টোরাইজেশন ব্যবহার করে ব্যাকট্র্যাকিং করে ক্রিপ্টো নাম্বারের সব ডিভাইজর জেনারেট করতে হবে। 1 থেকে 10^6 পর্যন্ত কোনো সংখ্যার মাক্সিমাম ডিভাইজর 240 টা। তাই প্রতি কুয়েরীতে জেনারেটেড ডিভাইজরের উপর লুপ চালানো সম্ভব। এই লুপ চালিয়ে প্রতি ডিভাইজরের কাউন্টের সামেশনই আউটপুট। Complexity: $O(Q(\text{sqrt}(N) + \text{Number Of Divisors}))$

Subtask 3

অনেকটা সাবটাস্ক 2 এর মতোই। মেইন কাজ হচ্ছে আরো এফিসিয়েন্টলি সব ডিভাইজর জেনারেট করা। লক্ষ্য করলে দেখা যায়, ক্রিপ্টো নাম্বারের কিউবরুট-এর বড় কোনো প্রাইম ডিভাইজর একটির বেশি থাকা সম্ভব না এবং সেটি ক্রিপ্টো নাম্বারের প্রাইম ফ্যাক্টোরাইজেশনে বর্গ আকারে থাকবে। এই প্রোপার্টি কাজে লাগিয়ে ক্রিপ্টো নাম্বারের কিউবরুট পর্যন্ত লুপ চালিয়ে প্রাইম ফ্যাক্টোরাইজ করতে হবে। বাকি কাজ সাবটাস্ক 2 এর মতোই। 1 থেকে 10^{12} পর্যন্ত কোনো ক্রিপ্টো নাম্বারের মাক্সিমাম 3500টার মত ডিভাইজর থাকে। তাই প্রতি কুয়েরীতে ডিভাইজরের উপর লুপ চালানো সম্ভব।

Complexity: $O(Q(N^{\frac{1}{3}} + \text{Number Of Divisors}))$

E. Easy

Setter: তৌহিদুল ইসলাম তনু

N টি সেট দেয়া থাকবে। তাদের মধ্যে থেকে কিছু সংখ্যক সেট এমনভাবে সিলেক্ট করতে হবে যাতে সিলেক্টেড সেটগুলোর মধ্যে কোন কমন উপাদান না থাকে এবং সিলেক্টেড সেটগুলোর উপাদান সংখ্যা ম্যাক্সিমাম পসিবল হয়।

Subtask

সেটগুলোর মধ্যে যতভাবে সম্ভব সেট সিলেক্ট করে দেখতে হবে তাদের উপাদানগুলো ইন্টারসেক্ট করতেছে কিনা , ইন্টারসেক্ট না করলে চুজেন সেটগুলোর টোটাল এলিমেন্ট সংখ্যা দিয়ে আউটপুটকে ম্যাক্সিমাইজ করতে হবে । কমপ্লেক্সিটি $O(2^N \times M)$, এখানে N টোটাল সেটের সংখ্যা এবং M টোটাল এলিমেন্টের সংখ্যা ।

Full Score

প্রথমে প্রত্যেক সেটের জন্য বের করতে হবে এই সেটের সাথে কোন কোন সেটের ইন্টারসেকশন আছে। $O(M \log M \times N)$ কমপ্লেক্সিটিতে এটা করা যাবে। এখানে M মানে সব সেট মিলিয়ে টোটাল এলিমেন্টের সংখ্যা এবং N মানে টোটাল সেটের সংখ্যা। এরপর বিটমাস্ক ডিপি করে আউটপুট বের করা যাবে, কমপ্লেক্সিটি $O(2^N \times N)$ ।

F. Hello Choto Bondhu!

Setter: শাহওয়াত হাসনাইন

একটি সংখ্যা n দেওয়া আছে। তোমাকে n এর বর্গের শেষ সংখ্যা টা বের করতে হবে। তবে, এখানে n অনেক বড় হতে পারে। এর মান সর্বোচ্চ হতে পারে 10^{100000} এর মতন।

Solution for Full Score

ধরো, তোমাকে একটি সংখ্যা n দেওয়া আছে। সংখ্যাটিকে লেখা যায় $(10m + k)$ আকারে। যেখানে, k হলো n এর শেষ ডিজিটটি।

$$\text{সুতরাং, } (10m + k)^2 = 100m^2 + 20mk + k^2 = 10(10m^2 + 2mk) + k^2$$

এখান থেকে আমরা দেখতে পারি, k যদি n এর শেষ ডিজিট হয় তবে n^2 এর শেষ ডিজিট অবশ্যই k^2 এর শেষ ডিজিট হবে।

তাই, এক্ষেত্রে আমাদের n এর শেষ ডিজিটের বর্গের শেষ সংখ্যাটা দেখলেই হবে।

পরিশেষঃ ফুল স্কোরের জন্য সংখ্যাটি এতো বড় দেওয়া থাকবে যে, তা `integer`, `long long integer` এ ইনপুট নেওয়া যাবে না। তাই, মূলত স্ট্রিং এ ইনপুট নিতে হবে।

G. The Depressed Guy

Setter: ধুব মিত্র

প্রথম এবং শেষ নাম্বার একই এরকম সবথেকে বড় সাব-অ্যারেটি বের করতে হবে।

Subtask 1

n^2 সলিউশন। অ্যারের উপর নেস্টেড লুপ চালিয়ে সব থেকে বড় সাব-অ্যারে যেখানে প্রথম আর শেষ নাম্বার একই সেটা বের করতে হবে।

Subtask 2

শুধু n এর একটি লুপ চলবে এবং প্রতিটি নাম্বার এর সর্বোচ্চ এবং সর্বনিম্ন ইন্ডেক্স এর ডিফারেন্স বের করতে হবে। এক্ষেত্রে নাম্বার এর জন্য অ্যারে ইউজ করা যেতে পারে।

Subtask 3

এখানে নাম্বারের কন্সট্রেন্ট বেশি তাই এখানে ম্যাপিং করে নিতে হবে।

H. Birthday Gifts

Setter: ফাহিম শাহরিয়ার স্বাক্ষর

N জন লোক কে N টা দোকানে পাঠাতে হবে। তারা প্রত্যেকে একটি নির্দিষ্ট পরিমাণ গিফট কিনে আনতে পারবে। প্রত্যেক লোক প্রতিটি পণ্য একবার করেই কিনতে পারবে। আবার, দুজন লোক একই পণ্য কিনতে পারবে না। এমনভাবে লোকগুলোকে দোকানে পাঠাতে হবে যাতে তারা সর্বোচ্চ সংখ্যক গিফট কিনতে পারে।

Subtask 1: Greedy

আমরা প্রথমে $N!$ উপায়ে কর্মচারী গুলোকে দোকানে পাঠাব। এরপর আমাদের এমনভাবে গিফটগুলো নিতে হবে যাতে তারা সর্বোচ্চ সংখ্যক নিতে পারে। $N = 2$ এর জন্য, আমরা গিফট গুলোকে 4 ভাগে ভাগ করে নিব।

- যেসব গিফট কেউই কিনতে পারবে না,
- যেসব প্রথম জন কিনতে পারবে,
- যেসব দ্বিতীয় জন করতে পারবে,
- যেসব দুজনই কিনতে পারবে।

যেসব গিফট একজন করে কিনতে পারে আমরা সেগুলো আগে কিনব। তারপর দুজনই কিনতে পারবে সেগুলো কিনবে।

Subtask 2: Flow

আমরা একটা ফ্লো গ্রাফ বানাবো। সোর্চ থেকে দোকান গুলোতে ক্যাপাসিটি যোগ করব। গ্রাফের ডান দিকে প্রতিটা আইটেমের জন্য একটা নোড থাকবে। দোকানগুলোর নোড থেকে আইটেমের নোডগুলোতে ইউনিট ক্যাপাসিটি এর ফ্লো যাবে।

Subtask 3: More Flow

M এর মান এখানে 100000 হতে পারে। তাই আগের আইডিয়াটা এই সাবটাস্কে *TLE* দিবে। আমরা ডানদিকে প্রতিটা আইটেমের জন্য নোড না নিয়ে, আমরা আইটেম গুলো কিছু টাইপ এ ভাগ করে নিব। কোন কোন দোকানে আইটেমটি আছে কি নেই, এর উপর ভিত্তি করে আমরা আইটেম গুলোকো আলাদা করে নিব। এতে আমাদের নোড লাগবে 2^5 টি। এখন আমরা ফ্লো ব্যবহার করতে পারি, যা টাইম লিমিটে পাস করবে।

I. Chow, Mo

Setter: রাফিদ বিন মোস্তফা

একটি ট্রি এর প্রতিটি নোডে কিছু রঙ আছে। একই রঙের যেকোন নোড থেকে অন্য নোডে যাওয়া যায়। আবার ট্রি এর প্যারেন্ট চাইল্ড এজ ধরেও হাটা যায় ট্রি তে। প্রদত্ত একটি নোড থেকে অন্য একটি প্রদত্ত নোডের শর্টেস্ট পাথ বের করতে হবে।

Subtask 1

ট্রি এর এজগুলার সাথে সাথে একই রঙের যেকোন নোড থেকে অন্য নোডে এজ দিয়ে গ্রাফ তৈরি করে Dijkstra অ্যালগোরিদম দিয়ে শর্টেস্ট পাথ বের করা যাবে।
কমপ্লেক্সিটিঃ $O((V^2 + E) \log_2 V)$ ।

Subtask 2

সবগুলো এজের ভ্যালু সমান। সুতরাং BFS অ্যালগোরিদম দিয়ে আগের মতন করেই সলভ করা যাবে। কিন্তু, নোডের সংখ্যা বেশি হওয়ায় একই রঙের ভেতরে এজ তৈরি করে নেয়া যাবে না। লক্ষ্যণীয় যে, একটি রঙের একটি নোড থেকে অন্য নোডের দূরত্ব যদি আপডেট করা হয়, তাহলে আর ঐ রঙের অন্য কোন নোডের জন্য একই রঙের নোডগুলার দূরত্ব আপডেট করা লাগবে না।
কমপ্লেক্সিটিঃ $O((V + E))$ ।

Subtask 3

সাবটাস্ক ২ এর মতন করেই Dijkstra অ্যালগোরিদম দিয়ে শর্টেস্ট পথ পাওয়া যাবে। কমপ্লেক্সিটিঃ $O((V + E) \log_2(V))$ ।

J. Circle of Death

Setter: রেদওয়ানুল হক সৌরভ

Solution

প্রথমে দেখা যাক কিভাবে N থেকে প্রিয় সংখ্যাটি বের করা যায়।
 প্রাথমিক ধারণাটা জোসেফাস প্রব্লেম থেকে নেয়া। এই প্রব্লেম এ N টি মার্বেল
 বৃত্তাকারে থাকে এবং প্রতি দ্বিতীয় মার্বেল কে ফেলা হয়, যতক্ষণ পর্যন্ত না শুধুমাত্র
 একজন মার্বেল বাকী থাকে। বাকি থাকা মার্বেল টি বৃত্তে কততম মার্বেল ছিল এটা
 বলাটাই জোসেফাস প্রব্লেম।

Solution (contd.)

প্রথমে জোসেফাস প্রব্লেম এর সল্যুশন(শুধুমাত্র একবার বাকি থাকা মার্বেল এর ইন্ডেক্স বের করা)ঃ

- যদি ১ টি মার্বেল থাকে তাহলে আসার ১
- যদি ২ টি মার্বেল থাকে তাহলে আসার ১
- যদি ৩ টি মার্বেল থাকে তাহলে আসার ৩
- যদি ৪ টি মার্বেল থাকে তাহলে আসার ১
- যদি ৫ টি মার্বেল থাকে তাহলে আসার ৩
- যদি ৬ টি মার্বেল থাকে তাহলে আসার ৫
- যদি ৭ টি মার্বেল থাকে তাহলে আসার ৭
- যদি ৮ টি মার্বেল থাকে তাহলে আসার ১
- যদি ৯ টি মার্বেল থাকে তাহলে আসার ৩
- যদি ১০ টি মার্বেল থাকে তাহলে আসার ৫

এখানে একটা প্যাটার্ন লক্ষ করা যায়, সেটা হচ্ছে ১, ১, ৩, ১, ৩, ৫, ৭, ১, ৩, ৫.....

Solution (contd.)

প্রথমত, এরা সবাই বিজোড় সংখ্যা এবং দ্বিতীয়ত, এরা ১ থেকে শুরু হয়, কিছু বিজোড় সংখ্যা এর পর আবার ১ থেকে শুরু হয়। ভালমত লক্ষ্য করলে দেখা যাবে যে, যেসব সংখ্যা ২ এর ঘাত তাদের জন্য ফলাফল ১। যেমন

১ ($= 2^0$) এর জন্য আঙ্গার ১, ২ ($= 2^1$) এর জন্য আঙ্গার ১, ৪ ($= 2^2$) এর জন্য আঙ্গার ১, ৮ ($= 2^3$) এর জন্য আঙ্গার ১

তাহলে জোসেফাস প্রব্লেম এর সমাধান হচ্ছে এমন যদি বৃত্তে N টা মার্বেল থাকে তাহলে N থেকে প্রথমে সর্বোচ্চ ২ এর ঘাত বা পাওয়ার বাদ দিতে হবে। এরপর যদি x বাকি থাকে তাহলে আঙ্গার হচ্ছে x তম বিজোড় সংখ্যা, বা $2^x + 1$ ।

Solution (contd.)

এখন, আমাদের প্রব্লেম এ আসা যাক। আমাদের প্রব্লেম এ বলা হয়েছে যে যদি প্রথম বার x_1 তম মার্বেল টি বাকি থাকে, তাহলে এর পরের বার x_1 টি মার্বেল নিয়ে খেলা হবে, এরপর যদি x_2 তম মার্বেল বাকি থাকে তাহলে x_2 টি মার্বেল নিয়ে খেলা হবে এভাবে চলবে যতক্ষণ না পর্যন্ত x_i টি মার্বেল নিয়ে খেলার ফলাফল x_i হয়। যেমন, ৩ টি মার্বেল নিয়ে খেলা শুরু করলে ৩ তম মার্বেল ই বাকি থাকে। ৭ টি মার্বেল নিয়ে খেললে ৭ তম মার্বেল কি বাকি থাকে, এরকম। এটা বের করার জন্য আমাদের জোসেফাস প্রব্লেম এর সূত্রে ফিরে যেতে হবে।

Solution (contd.)

কোন সংখ্যা থেকে ২ এর সর্বোচ্চ ঘাত বিয়োগ করা মানে ওই সংখ্যা এর বাইনারী ফরম এ এর সবচেয়ে ডান দিকের ১ কে ০ করে দেয়া। যেমনঃ

৬ থেকে ২ এর সর্বোচ্চ ঘাত বের করে দেয়া হলঃ $6-8=2$

এখানে ৬ এর বাইনারীঃ ১১০

২ এর বাইনারীঃ ০১০

এবং কোন সংখ্যা কে ২ দিয়ে গুন করা মানে তার বাইনারী ফরম এ লেফট শিফট অপারেশন করা। এতে সবচেয়ে বাম এর বিট এ ০ আসে।

যেমনঃ $6*2 = 12$

৬ এর বাইনারী ঃ ১১০

১২ এর বাইনারীঃ ১১০০

আর ২ গুন করে ১ যোগ করলে সবশেষ এর বিট এ ১ যোগ হয়। তাহলে শেষ এ ০ এর জায়গায় অতিরিক্ত ১ আসবে।

তাহলে, যদি বৃত্তে ৫ (১০১ বাইনারি) টি মার্বেল থাকে তাহলে

১০১ \rightarrow ০০১ \rightarrow ০১০ \rightarrow ০১১ (৩)

Solution (contd.)

আমরা দেখতে পাচ্ছি যে যদি জোসেফাস প্রব্লেম এ একটা N দেয়া থাকে তাহলে তার বাইনারী তে সার্কুলার লেফট শিফট অপারেশন করলেই তার উত্তর পাওয়া সম্ভব। সার্কুলার লেফট শিফট মানে সব বিট একঘর বাম এ আসবে এবং সাথে সাথে সবচেয়ে ডান এর বিট টি সবচেয়ে বাম এর বিট হবে।

এখন যদি বার বার সার্কুলার লেফট শিফট করা হয় একটা সংখ্যা কে তাহলে শুধুমাত্র বাইনারি তে ১ গুলো বাকি থাকে, এবং ০ গুলি বাদ চলে যায়। যেমন ১০ এর বাইনারি ১০১০

১০১০ \rightarrow ০১০১ (১০১) \rightarrow ০১১ (১১) \rightarrow ১১

যদি N এর বাইনারি তে x টি ১ থাকে তাহলে, বার বার লেফট শিফট করলে একসময় $2^x - 1$ আসবে, এবং এরপরে আর কোন পরিবর্তন হবে না। আমাদের প্রব্লেম অনুযায়ী এটাই সেই প্রিয় সংখ্যা।

Solution (contd.)

কোন সংখ্যা এর বাইনারি তে কতগুলো ১ আছে সেটাকে ওই সংখ্যা এর পপকাউন্ট (popcount) ও বলে।

এখন ১ থেকে $2^k - 1$ এর ভিতরের সবগুলো সংখ্যা কে k টি বিট দিয়ে প্রকাশ করা যায় বাইনারি তে। এখন যেসব সংখ্যা এর পপকাউন্ট N এর পপকাউন্ট এর সমান, সেসব সংখ্যা বের করতে হবে। এর জন্য সবগুলো সংখ্যা খুজে দেখার দরকার নেই। k টি বিট এর ভিতরে x টি বিট নেয়া যায় kC_x ভাবে। এটাই উত্তর। ছোট করে বললে, N এর বাইনারি তে যে কয়টা ১ আছে এটা বের করতে হবে। এটা লুপ চালিয়ে অথবা c/cpp তে `__builtin_popcount` ব্যবহার করে করা যায়। ধরা যাক এটি x , এরপর kC_x হচ্ছে আঙ্গার।

K. A Girl Has No Name

Setter: রেজাউল হক

এখানে একটি রেঞ্জ দেয়া আছে L থেকে R । L থেকে R এর মধ্যে প্রতি নম্বরের জন্য একটি স্কোর ক্যালকুলেট করতে হবে। ধরা যাক একটি নম্বর হচ্ছে N , A হচ্ছে N এ যতগুলো অশূন্য ডিজিট আছে সেই সংখ্যা, আর B হচ্ছে সেগুলোর মধ্যে যতগুলো N কে নিঃশেষে ভাগ করে। এখন প্রতি নম্বরের জন্য স্কোর হচ্ছে B^A ।

Solution

ব্রুট ফোর্স সলিউশন লিখে ২০ পয়েন্ট পাওয়া যাবে। ১০০ পয়েন্টের জন্য ডিজিট ডিপি করতে হবে। ডাইনামিক প্রোগ্রামিং এর প্যারামিটার হিসেবে এলসিএম (লঘিষ্ঠ সাধারণ গুণিতক) রাখতে হবে, কারণ যে নম্বরগুলো দিয়ে ভাগ করতে হবে সেগুলোর একটা সুপারসেট হিসেবে লসাগু ব্যবহৃত হবে। এছাড়াও প্যারামিটার হিসেবে রাখতে হবে একটি মাস্ক যার সাহায্যে আমরা বুঝবো একটা স্টেটে এ পর্যন্ত কোন কোন ডিজিট ব্যবহার করা হয়েছে।

Solution (contd.)

আমরা জানি, ডিজিট ডিপি সলিউশনের ক্ষেত্রে আমরা রেঞ্জের ভিতর আমাদের কাক্ষিত যতগুলো নম্বর আছে সেগুলো বানানোর চেষ্টা করি। প্রতি স্টেট এ আমাদের লক্ষ্য থাকে কোন একটি নির্দিষ্ট ডিজিট এখানে append করলে আমরা রেঞ্জের ভিতরে আছি কিনা।

`dp(index, isSmall, mask, remainder)`

- index - কোন ডিজিটে আছি
- isSmall - যে নম্বরটি বানাচ্ছি সেটি ইতোমধ্যে ছোট হয়ে গেছে কিনা
- mask - একটি ইন্টিজার যেটি দেখে বুঝা যাবে এ পর্যন্ত কোন কোন ডিজিট নিয়েছি
- remainder - লসাগু দ্বারা ভাগ করার পর যে ভাগশেষ থাকে সেটা