সম া ও সমাধান

বই - ১

মাঃ মাহবুবুল হাসান

সে র, ২০১৭

১

২

সূচীপ

১ একটুস খািন কথা ৯

২ সহজ িকছু সম া ১৩ ২.১ অ শীলনী . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ২৩ ২.১.১ সম া . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ২৩ ২.১.২ িহ ট . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ২৩

৩ STL াকিটস ২৫ ৩.১ অ শীলনী . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ৩৪ ৩.১.১ সম া . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ৩৪ ৩.১.২ িহ ট . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ৩৪

৪ Mathematics স িক ত সম া ৩৭ ৪.১ অ শীলনী . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ৬৯ ৪.১.১ সম া . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ৬৯

৫ Bruteforce এবং Backtrack স িক ত সম া ৭১ ৫.১ অ শীলনী . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ৭৮ ৫.১.১ সম া . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ৭৮ ৫.১.২ িহ ট . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ৭৮

৬ Data Structure স িক ত সম া ৮১ ৬.১ অ শীলনী . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ৯৪ ৬.১.১ সম া . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ৯৪ ৬.১.২ িহ ট . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ৯৫

৭ Greedy ৯৭ ৭.১ অ শীলনী . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ১০৬ ৭.১.১ সম া . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ১০৬

৮ Dynamic Programming ১০৭ ৮.১ অ শীলনী . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ১৩০ ৮.১.১ সম া . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ১৩০ ৮.১.২ িহ ট . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ১৩১

৯ Graph Theory ১৩৩ ৯.১ অ শীলনী . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ১৪৭ ৯.১.১ সম া . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ১৪৭

৩

১০ Adhoc ১৪৯ ১০.১ অ শীলনী . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ১৬৪ ১০.১.১ সম া . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ১৬৪

১১ Geometry ১৬৫ ১১.১ অ শীলনী . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ১৭৭ ১১.১.১ সম া . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ১৭৭

৪

নকশা তািলকা

৪.১ Pascal Triangle . . . . . . . . . . . . . . . . . . . . . . . . . . . . ৪২ ৪.২ UVa 1638 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ৪৮ ৪.৩ (*a, b*) = (4*,*3) এবং *S* = 4 . . . . . . . . . . . . . . . . . . . . . . . ৫১ ৪.৪ Ellipse এর উপর *n* = 9 িট িব . . . . . . . . . . . . . . . . . . . . . ৫৬

৫.১ UVa 1343 এর বাড . . . . . . . . . . . . . . . . . . . . . . . . . . ৭৩

৬.১ একিট tree এর Euler path . . . . . . . . . . . . . . . . . . . . . . . ৯৪

১১.১ Latitude এবং Longitude, wikipedia হেত . . . . . . . . . . . . . . . ১৭৫

৫

৬

সারণী তািলকা

৪.১ 30/i এর টিবল . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ৫৮

৭

৮

অধ ায় ১

একটুস খািন কথা

া ািমং কে ট এর ইিতহাস িক অেনক পুেরােনা। তামরা িক কউ ACM ICPC World Finals এর prize giving ceremony দেখছ? এখন তা আবার ই টারেনেটর কল ােন সই অ ান দখেত সশরীের হািজর হেত হয় না। তা যা বলিছলাম, সই অ ােন িতবার িবল পাউচার িত বছেরর িবজয়ীেদর নাম বেল আর সই িল হয় 1977 সাল থেক। তার মােন আজ থেক ায় 40 বছেররও আেগ া ািমং কে ট হেয়েছ। আজ 2017 সােল এেস া ািমং কে ট এর িবিভ platform দািড়েয় গেছ। ACM ICPC World Finals, IOI, Codejam, Hackercup, Codefestival, Codechef Snackdown এবং আরও অেনক onsite event হয়। এছাড়াও online এ া ািমং িতেযািগতা করার এবং াি স করার ব সাইট আেছ। এছাড়াও আেছ া ািমং িতেযািগতা িনেয় আেলাচনা করার অেনক ফারাম। গত 4 বছেররও কম সমেয় া ািমং িতেযািগতার য সার হেয়েছ তা আসেল কথায় কাশ করা যােব না। িক কে র িবষয় হল আমােদর দেশ স তুলনায় তমন সার নই। আমােদর দেশ গত দশ পেনেরা বছর ধের গেড় িত বছর চার পাঁচিট কের কে ট হয়। িকছু িদন আেগ থেক (2011 থেক ) িত বছর া ািমং ক া করার চ া করা হয়। এছাড়াও ুল কেলজ লেভেল NHSPC আয়জন করা হে অ িকছু িদন আেগ থেক। এেতই সীমাব আমােদর া ািমং কে ট । সই তুলনায় রািশয়া, চীন, পাল া ড যারা িকনা া ািমং কে টে সবসময় এিগেয় থােক তােদর আেয়াজন নেল বা দখেল আমার বশ খারাপ বা আফসস লােগ। এজ না য তারা অেনক িকছু কের। বরং তারা আমােদর থেক অেনক কম onsite কে ট কের। তারা যা কের তাহল quality পূণ ক া । সখােন তােদর বুেড়া কে টে টরা এেস quality পূণ াস নয় এবং quality পূণ বেলম িনেয় আেলাচনা কের। ফেল দখা যায় আমরা আমােদর দেশর কে টে 10 িট সম ার ভতের 7-8 িট কের সমাধান করেত পারেলও তােদর কে টে য 10 িট সম া থােক তার 2 বা 3 িট টেনটুেন আমরা সমাধান করেত পাির আর তারা িঠকই 8-9 িট কের সমাধান কের বেস থােক। বলার অেপ া রােখ না আমােদর আর তােদর ভতের আকাশ আর পাতাল তফাৎ।

আমার কােছ মেন হেয়েছ সম া সমাধােনর িট হািতয়ার আেছ। এক ান, ই চ া। তুিম যিদ না জান bfs িক বা dp িক তাহেল তা তা স িক ত সম া সমাধান করা কিঠন। হয়েতা bfs বা dp আসেল বশ সহজ িজিনস, কউ বই না পেড়ই িনেজ িনেজ িশখেত পারেব বা িনেজ থেকই সসব সমাধান বর করেত পারেব। িক fft বা kmp বা convex hull ইত ািদ স িক ত algorithm বা theory এসব যিদ তুিম না পড় না িশখ না জান তাহেল সসব স িক ত সম া সমাধান করা খুবই কিঠন হেব। সজ থমত ান দরকার। আর ি তীয় িজিনস হল চ া। তামােক সম ায় বলা থাকেব না য এই সম া bfs িদেয় সমাধান হেব। তামােক িচ া কের, নানা modify কের বুঝেত হেব য এটা এভােব কের bfs করেল সমাধান হেব।

িকছু ান যমন bfs, dfs, mst, kmp, dp, gcd ইত ািদ খুবই সহজ িজিনস। যারা এসব এখনও পার না তারা হয়েতা আমােক মেন মেন গািল িদ য কত কিঠন িজিনস, আর আিম িকনা এেদর খুবই সহজ বলিছ। িক সিত বলেত া ািমং কে টে আরও অেনক কিঠন কিঠন িজিনস আেছ। সসেবর

৯

তুলনায় এসব খুবই সহজ।আমার মূল ল িছল সসব কিঠন িজিনস িনেয় বই িলখার। যমন-fft, nft, persistent data structure, link cut tree ইত ািদ। কারণ আমার মেন হেয়েছ সহজ িবষয় িল তা এক জন চাইেলই নট থেক পেড় িশখেত পারেব িক এই কিঠন িবষয় িল ইংেরিজেতই ভাল কের নাই। ভাল কের কাথাও থাকেলও এক জায়গায় সব নই। তাই বাংলায় িলখেল হয়েতা যারা বাংলােদেশর advanced contestant তােদর জ ভাল হত। িক সই বই িলখেত িগেয় মেন হেয়েছ য আসেল সহজ িবষেয়র উপেরও বই দরকার। এজ না য আমােদর দেশ advanced contestant এর সংখ া খুবই কম, বরং এজ য আিম যখন advanced িবষেয়র উপর িলখব তখন যন এটা মেন না হয়, আের এই িবষয় িক ওরা জােন? অমুক টকিনক িক ওরা জােন? এসব ভেবই আমার " া ািমং কে ট - ডটা াকচার ও অ ালগিরদম" বই িলখা। বইটা িলখা হয় 2011 সােল আর বিশর ভাগ িজিনস িলখা শষ হয় 2013 সােল। এরপর নানা রকম আলেসিম পার কের 2015 ত তা pdf আকাের ছাড়া হয় এবং 2016 ত হাড কিপ িহসােব। কিঠন িবষেয়র উপর বই িলখার ই া িনেয় কের 5 বছর কািটেয় িদেয়িছ সহজ িবষেয়র উপর িলেখ। িক সই বই এ ধু theory িছল। সখােন তমন কােনা সম া িনেয় আেলাচনা িছল না, ে েমর িল ও িছল না। সিত বলেত, ঐ বইএ আরও একিট চ া ার িছল যখােন ায় 300-400 সম া িনেয় আেলাচনা করা িছল (যত দূর স ব 2013 সােলর সব icpc regional এর problem এবং solution িনেয় িছল, এমনিক neerc, ceerc এবং chinese regional িলও িছল)। িক বই এর আকার এেতা িবশাল হি ল আর সই সম ার আেলাচনা এেতা সংে েপ িছল য আিম পের ঐ অংশ মুেছ ফিল (এমিনেতই theory আেলাচনা অংশ সংি ও বেল যই পিরমান গালাগািল হজম করেত হেয়েছ স তুলনায় ঐ চ া ার মুেছ িদেয় ভালই হেয়েছ ব িক!)। থম বইএ থম হািতয়ার ( ান) িনেয় কথা হেলও ি তীয় হািতয়ােরর ( চ া) কােনা িকছুই িছল না। সই জ ই আমার এই ি তীয় বই। আিম িনেজ বেলম ক াটাগরাইেজসেন ওত অভ না। তাই এই বইেয় যত সম া আেলাচনা করা হেয়েছ তার িল িল আিম িবিভ সাস হেত িনেয়িছ। মূল িল হল িজয়া িলউ এর এক ন র বই (সহজ বই) এর িল । এছাড়াও আিম timus এর beginner ে েমর একিট িল ব বহার কেরিছ। LightOJ এর টিপক এর িল হেতও বশ িকছু সম া আিম ব বহার কেরিছ। জ ািমিতর জ বাংলােদেশর জ ািমিতর নািফ এর একটা বেলম িল িছল আমােদর বুেয়েটর েপ সই িল ব বহার কেরিছ। Greedy সম ার জ codeforces এর িকছু সম া ব বহার কেরিছ। এরকম নানা জায়গা হেত বেলম িনেয় সসব িনেয় আেলাচনা করা হেয়েছ।

এই বইেয়র মূল উে যিদ তুিম মেন কর িবিভ টিপেক াি স করা তাহেল ভুল ভাবেব। মূল উে হল, চীেনর বা া কে টে টরা িজয়া িলউ এর যই বই পেড় তােত আেলািচত বেলম েলা তামােদর সামেন তুেল ধরা। আর বুঝােনা কন তারা এেতা ভাল পাের আর কন আমরা পাির না। তােদর সহজ বইেয়ই যিদ এেতা কিঠন বেলম থােক তাহেল কিঠন বইেয় না িক আেছ! যিদ তামরা সিত কােরর ভাল কে টে ট হেত চাও, তাহেল এই বইেয়র বেলম িলেক তামােদর কােছ সাজা লাগেত হেব। আর যারা শেখর বেশ কে ট করেত চায়, খাটা খাটিন করেত চায় না, তারা এই বইেক িনরাপদ রে দূের সিরেয় রােখা।

সাধারনত বইেয় িতিট অধ ােয় বেলম িল তােদর difficulty অ যায়ী আেছ। তরাং তুিম একিট কের বেলম দখ আর িনেজরা তা সমাধান করার চ া কর। যিদ িনেজরা পার তাহেল তা ভাল কথা, আর না পারেল সমাধান একটু একটু কের পেড় দখ। এক লাইন ই লাইন কের পড় আর িনেজরা আবার িচ া কর। যিদ িনেজ সমাধান করেত পার তাহেলও এই বইেয়র সমাধান পড়। কারণ দখা যােব, যিদও এই সম া O(*n*3) এ সমাধান হয় িক এর O(*n*2) সমাধান হয়েতা আিম বণ না কেরিছ। বা অেনক সময় আিম নানা ছাট খাট ি েকর কথা বেলিছ তা হয়েতা অ ে েম কােজ আসেব।

অেনক সময় এটাও হেত পাের য তুিম কােনা একিট সমাধান বুঝছ না। এরকম না হওয়াটাই অ াভািবক (সংি ও সমাধােনর জ ই হাক বা কিঠন সম ার জে ই হাক)। এখােন অেনক সম া আেছ য িল আমার িনেজর সমাধান করেত বা অে র সমাধান পেড় বুঝেত 4-5 িদন কের লেগেছ। তরাং তুিম যিদ 1 িদন ব য় কেরই হাল ছেড় দাও তাহেল হেব না। কাগেজ কলেম িবিভ কস িনেয় িচ া করেত হেব আিম িক বলিছ, বা তা কীভােব হে বা হেত পাের।

অেনেকই বেল বই আরও সহজ কের িলখেত, িক সিত বলেত এর থেকও সহজ কের িলখার আমার সময় নই। আিম যিদ আরও সহজ কের আরও অেনক সময় িনেয় িলিখ তাহেল আমার আসল

১০

উে (কিঠন িবষয় িল শখা এবং তা িনেয় লখা) আমার কখনই পুরন হেব না। বা যারা বত মােন advanced coder আেছ তারা কখনও কিঠন িজিনস িল িশখেত পারেব না। সহজ িবষয় িলর উপর িলখা বা শখােনার অেনেকই আেছ িক কিঠন িজিনস িল িনেয় সংি িলখাও নই। তরাং আিম যিদ কখনও আমার ল পুরন করেত পাির তাহেল হয়েতা কােনািদন আবার এই "কিঠন" কের িলখা বই িরভাইজ কের সহজ কের আরও ছিব িদেয় িলখব। িক আশা কির তত িদেন আমােদর দেশ এেতা 2400+ রিটং এর কাডার থাকেব য "এই কিঠন বই বুঝিছনা" এ কথা বলা মা েষর পিরমান কেম যােব বা তােদর মধ হেত আর কউ সহজ কের বই িলখেব।

আেগই বেলিছ এই বইেয় UVa, Timus, Codeforces, LightOJ থেক সম া আেছ। আিম ে েমর না ােরর পাশাপািশ ে েমর নাম িদেয়িছ যােত তামােদর খুঁেজ পেত িবধা হয়। িবেশষ কের codeforces এর ে এই সম া বিশ দখা যায় কারণ সখােন একিট ে েমর না ার দওয়া বশ কিঠন। আর মােঝ মােঝ আিম বেলম statement একটু পিরবত ন কের িনেয়িছ। যমন দখা যােব মূল ে েম বলা হেয়েছ য ছাগেলর কািহনী আিম িলেখিছ গাধার কািহনী। অেনক সময় মূল ে েম বেলেছ য অমুক অমুক িডিজট ব ািতত অ সকল িডিজট ব বহার করা যােব। িক আিম ে েম িলেখিছ য অমুক অমুক িডিজট কবল ব বহার করা যােব। এসেবর মেধ িক তমন

কােনা তফাৎ নই। তরাং আশা কির তামােদর বুঝেত সম া হেব না।

আর আমরা এই বইেয় আেলািচত সম া িলর কাড রাখার জ একিট git repository খুেলিছ যার িলংক (https://github.com/shanto86/problem-book-1-solutions). তামরা চাইেল তামােদর ac কাড এখােন পাঠােত পার যােত অ রা তা দেখ িশখেত পাের। আমরা চ া করব কােনা একিট সম ার একিট মা সমাধান এখােন রাখার। তেব যিদ কউ স ূণ িভ ভােব কা- না সম া সমাধান কের তাহেল আমরা সই নতুন সমাধানও রাখেত পাির। সই সােথ ধ বাদ Nayeem Jahan Rafi, Evan Hossain, Rezwan Arefin, Md Sahedul Islam Sohel, Mohtasim Bellah, Moudud Khan Shahriar, Abdullah Al Raqibul Islam, Imran Ziad এবং Sumit Saha ক যারা বইেয়র টকিনকাল িরিভউ করেত সাহায কেরেছ।

১১

১২

অধ ায় ২

সহজ িকছু সম া

এই অধ ােয় আমরা সহজ িকছু সম া িনেয় আেলাচনা করব।আশাকির এই অধ ােয়র সম া িল সমাধান করেত কােনা algorithm লাগেব না। তামােদর online judge (oj) আর c/c++ এর সােথ পিরিচত করাই এই অধ ােয়র উে । তামােদর যারা ইেতামেধ ই online judge এ অ ত ১০০ িট সম া সমাধান কের ফেলছ তােদর জ হয়েতা এই অধ ায় ওত কােজ আসেব না। তামরা বরং ঘিড় ধের এই অধ ােয়র সম া িল সমাধান করার চ া কের দখেত পার। কত কম সমেয় কত কম পনাি েত সব সমাধান করা যায় স চ া কর। আর সব সমাধান শেষ একবার আমার বলা সমাধান িলেত চাখ বুিলেয় িনেত পার। হয়েতা িকছু নতুন িটপস পেয় যেত পােরা!

অ শীলনী বশ অেনক িল সম া দওয়া আেছ। সব িলই মুলত c/c++ জানা থাকেলই হেব। তরাং তামরা যারা া ািমং কে ট জগেত নতুন তােদর জ ই মুলত এই অধ ায়। আেরকটা িজিনস, িকছু িকছু সম া আেছ য েলা হয়েতা মেন হেব কাড করা পইন। িক আমার মেত তাও সসব সম া করা উিচত। এরকম সম া তামােদর debug করার মতা বৃি কের এবং র কের কাড করা িশখেত সাহায কের। তরাং কােনা সম া দখেত পইনফুল হেল তােক পাশ কািটেয় যও না। আর কােনা একিট সম া সমাধান করা হেল তামরা চাইেল নেট সাচ কের অ েদর কাড দখেত পার। এেত কের নতুন টকিনক িশখেত পারেব।

Timus 1000 A+B Problem

**সমস া:** তামােক a এবং b িট সংখ া দওয়া আেছ। তােদর যাগফল ি ট করেত হেব।

**সমাধান:** খুবই সহজ সম া। তেব সরাসির কাড করেত যাবার আেগ দেখ নাও a আর b এর িলিমট কত। ঃেখর িবষয় হেলা এই সম ায় এেদর কােনা িলিমট দওয়া নই। আেগর যুেগ এরকম সম া ায়ই হেতা। অথ াৎ দখা যায় ে েম ইনপুট ভ ািরেয়বেলর িলিমট িল দয়া নই। যিদ এটা কে টে হয় তাহেল judge ক clarification এর মাধ েম িজ াসা কের নাও। আর যিদ এটা offline হয় অথ াৎ াকিটেসর জ online judge এ সমাধান করার সময় এরকম হেয় থােক তাহেল েপ বা ফারােম িজ াসা কর অথবা ফারােমর পুরেনা পা একটু ঘঁেট দেখা। এরপরও যিদ না বুঝেত পােরা কী কািহনী তাহেল আমার মেত সই সম া না চ া করাই ভাল। OJ িলেত অেনক ভাল ভাল সম া থােক, ধু ধু একটা ambiguous সম ার পছেন সময় ব য় করার কােনা মােন নই।যারা কবলমা Cিশখছতারা অেনেকই ইনপুট নবার আেগ promptি ট কের থাক। Prompt হেলা "please enter your number", "please input a" ইত ািদ। বলেত ি ধা নই আিম যখন আমার জীবেনর থম সম া সমাধান কেরিছলাম (UVa 100) তখন এই কাজ কেরিছলাম। এসব prompt দবার আসেল কােনা দরকার নই। বরং এসব prompt িদেল তুিম wrong answer (বা কখনও কখনও output limit exceed বা presentation error) পাবা। কন? কারণটা বাঝার

১৩

জ তামােক আেগ িচ া করেত হেব online judge কীভােব কাজ কের। Online judge তামার কাড নয়, াভািবক ভােবই তামার কাড compile কের এবং সব শেষ তার কােছ থাকা এক বা একািধক input ফাইল তামার কাডেক দয়। তামার কাড সই ইনপুট িল standard input হেত পেড় (অথ াৎ সাধারণ scanf, cin ইত ািদ িদেয়)। কােনা কােনা oj ত অব আমােদর ফাইল হেত পড়েত হয় (fscanf, fopen, freopen ইত ািদ ব বহার করেত হয় সে ে )। কান জােজ কীভােব ইনপুট আউটপুট করেত হেব তা সাধারনত তােদর frequently asked question বা এই জাতীয় সকশেন িলখা থােক। যাই হাক, ক াইল করা শেষ তামার কাড সই সব ইনপুট পেড় output দয়। এবার তামার দয়া এই output তারা তােদর কােছ থাকা সিঠক output এর সােথ মলায়। এখন তুিম যিদ অিতির prompt দাও সটা তা আর জাজেদর output এ থাকেব না তাই না? আর তাছাড়া তুিম ভেব দেখা যিদ তুিম ইনপুট নবার জ prompt দাও তাহেল তামার আসল উ র আর prompt আলাদা করার িক কােনা উপায় আেছ? নই। কারণ জাজ তা সব আউটপুট িনেয় এর পর চক কের। স তা জােন না কানটা prompt আর কানটা আউটপুট। সজ prompt দবার কােনা দরকার নই। আর াঁ, আিম জািন য আিম যত সহেজ oj কাজ কের বললাম, আসেল তা অেতা সহজ না, িক মাটােমািট ঐরকম ভােবই কাজ কের।

অেনক কথা হেলা, আমােদর এই সম ায় িফের আসা যাক। তুিম এই সম ার ফারােম একটু ঘাঁটেলই বুঝেব য এখােন a আর b ক int এ িনেলই চলেব। তাহেল আর দরী কন, ঝটপট কাড কের ফল। ৬-৭ লাইেনর বিশ কাড হওয়া উিচত না!

Timus 1001 Reverse Root

**সমস া:** সেব া 1018 হেত পাের এমন িকছু অঋণা ক পূণ সংখ া দয়া আেছ। তামােক তােদর square root িল (সেব া 4 decimal digit পয ) উলেটা অড াের ি ট করেত হেব। অথ াৎ থম সংখ ার square root সবেশেষ, ি তীয় সংখ ার square root শষ হেত ি তীয়েত এরকম কের ইনপুেটর সব শেষর সংখ ার square root আউটপুেট সবার থেম থাকেব। ইনপুট এর সাইজ সেব া 256KB. উদাহরণ প- তামার ইনপুেটর সংখ া িল যিদ হয় 1, 4 তাহেল আউটপুট হেব 2, 1.

**সমাধান:** এিটও বশ সহজ। তেব একটা সম া হেলা কত িল না ার দয়া থাকেব তা বলা নই। ধু বলা আেছ ইনপুেটর সাইজ 256KB. আমরা জািন এক character এ 1 byte১. তাহেল ইনপুেট থাকেব 256K টা িডিজট। আমরা আমােদর িহসােবর িবধার জ ধের িনেতই পাির য এেককটা িডিজট এেককিট সংখ া। যিদও আসেল তা না, এেদর মােঝ অেনক space থাকেব। িক আমােদর তা exact সংখ া দরকার নই। মাটােমািট একটা estimate হেলই হেব। তাহেল 256K সাইেজর একিট অ াের িনেলই িনঃসে েহ যেথ হেব ২। এই অ ােরেত তুিম ইনপুট িনেয় রােখা। ইনপুট শেষ তুিম এই অ ােরর পছন থেক এেস এেক এেক সবার square root ি ট কর। আর আশা কির ইেতামেধ ই খয়াল কেরছ য আমােদর int ব বহার করেল চলেব না, long long ব বহার করেত হেব বা চাইেল double এও ইনপুট িনেত পােরা। কারণ আমােদর ইনপুেটর সংখ ার িলিমট 1018.

আরও এ েনার আেগ সাইেজর ব াপারটা আেরকটু পির ার কের নওয়া যাক। এইেয এখােন 256K সাইেজর একিট long long এর অ াের িনেল, তার মােন এই অ ােরিট memory ত কত জায়গা দখল কের? তার আেগ জানা দরকার একিট long long মমিরেত কত জায়গা দখল কের। আশা কির তামােদর মেন আেছ int নয় 32 িবট বা 4 বাইট আর long long নয় 64 িবট বা 8 বাইট (1 বাইট = 8 িবট)। এর মােন আমােদর মাট 8 *×* 256 = 23 *×* 28 = 211 িকেলাবাইট বা 2 মগাবাইট মমিরর েয়াজন হেব। float বা double কত সাইজ দখল কের? এটা মেন রাখা সহজ। float নয় int এর সমান আর double নয় long long এর সমান। যিদও compiler বা os ভেদ কােনা একিট ডটা টাইপ এেকক compiler বা os এ িভ িভ সাইজ দখল করেত পাের। িক সটা িনেয় আসেল িচ া না করেলও চলেব। তাহেল এই িহসাব থেক তামরা জানেল য তামােদর ায় 2 মগাবাইট এর মত জায়গা লাগেব। এিট কন পূণ ? কারণ িতিট সম ায়

১আেগ না জানেল এখন জানেল। যেকােনা একিট character ইনপুেট বা text ফাইেল 1 বাইট জায়গা দখল কের। ২এখােন K(Kilo) এর মােন হেলা 210, 1000 না। একই ভােব M(Mega) হেলা 220 আর G(Giga) হেলা 230.

১৪

time limit এর পাশাপািশ memory limit ও দয়া থােক। যমন এই সম ার memory limit হেলা 64 মগাবাইট। খয়াল করেব তামার কাড যন এর বিশ মেমাির ব বহার না কের। না হেল memory limit exceed খেয় যেত পােরা! বিশর ভাগ সময়ই এই িলিমট ল করার দরকার হয় না। িক মােঝ মােঝ দখা যায় তুিম DP করেত চা িক DP এর অ াের অেনক বিশ মমির দখল কের বেস আেছ। সসব সময় তামােক মমির অি মাইজ করেত হেত পাের।

এই সম ার ে তামরা চাইেল অ ােরর সাইজ কত িনেত হেব সই িহসােবর ঝােমলায় না িগ- য় stl এর vector ব বহার করেত পােরা। একিট কের ইনপুট নাও আর vector এ push back কর। সবেশেষ vector এর পছন থেক আেসা। iterator ব বহার করার কানই দরকার নই (iterator িক যিদ না জান তাহেল আপাতত জানার দরকার নই, দরকােরর সময় িশেখ িনেলই হেব)। আিম মেন হয়না জীবেন vector এর জ iterator ব বহার কেরিছ। তামােদর িবধার জ vector ব বহার কের এর কাড ২.১ এ দওয়া হেলা।

িফিরি ২.১: timus1001.cpp ১ #include *<*stdio.h*>* ২ #include *<*math.h*>* ৩ #include *<*vector*>* ৪ using namespace std; ৫৬ int main() *{* ৭ vector*<*long long*>* V; ৮ long long a; ৯ while (scanf("%lld" &a) != EOF) *{* ১০ V.push back(a); ১১ *}* ১২ for (int i = V.size() *−* 1; i *>*= 0; i*−−*) *{* ১৩ printf("%.4lf*\*n", sqrt(V[i])); ১৪ *}* ১৫ return 0; ১৬ *}*

এখােন আেরকটা িজিনস। সটা হল EOF. খয়াল কর আমােদর ইনপুেট যত ণ সংখ া আেছ তত ণ আমােদর ইনপুট িনেত হেব। এই কাজ আমরা এই EOF এর মাধ েম করেত পাির। যখন ইনপুট নওয়া শষ হেয় যায় তখন তুিম মেন করেত পার য scanf ফাংশন EOF (End Of File) িরটান কের। তরাং আমরা তত ণ ইনপুট েসস করব যত ণ না scanf ফাংশন EOF িরটান কের।

UVa 12108 Extraordinarily Tired Students

**সমস া:** মেন কর একিট ােস সেব া দশ জন ছা ছা ী আেছ। তারা েত েকই ঘুমাে । অব তােদর ঘুমােনার িনজ একিট াইল আেছ। েত েকর াইল আলাদা হেত পাের। তারা a সময় ঘুমায়, b সময় জােগ, a সময় ঘুমায়, b সময় জােগ- এভােব চলেত থােক (a ও b সেব া 5 হেত পাের)। তােদর থম অব া 1 হেত a + b এর মােঝ যেকােনািট হেত পাের। যমন- a = 2 আর b = 3 হেল এবং তােদর থম অব া 3 হেল তারা 2 সময় জেগ থাকেব (3 এ আেছ মােন 2 টা ঘুম ইেতামেধ ই শষ এবং 1 টা জাগাও শষ), পেরর 2 সময় ঘুমােব, 3 সময় জাগেব, 2 সময় ঘুমােব, 3 সময় জাগেব- এভােব চলেব। সবার জ a ও b দওয়া থাকেব আর সই সােথ দয়া থাকেব র অব া। বলেত হেব থম কান সমেয় তারা সবাই জেগ থাকেব। যিদ এরকম সময় পাওয়া স ব না হয় তাহেল -1 ি ট কর।

১৫

**সমাধান:** যারা গিণত পছ কর তােদরেক বলিছ- তামােদর trade-off িশখেত হেব। Trade-off মােন হেলা কােনা একিট সম া পাবার পর সটা গিণত িদেয়ই সমাধান করবা নািক িকছু র গািণিতক ভােব কের বািকটু কি উটােরর উপর ছেড় িদবা- সটা। যমন এই সম ার ে কউ যিদ মেন কর আের এটা তা মজার গািণিতক সম া লাগেছ, লসা লসা গ ! কথা িঠক, িক তাই বেল পুেরাটা গািণিতক ভােব করা মেন হয় বশ মুশিকল হেব। িতিট ছা ছা ীর ঘুেমর সাইেকল খুব জাড় 10 (a+b). মােন িত *a* + *b* সময় পর পর একই অব া িফের আসেব। আমােদর সম ায় সেব া দশজন থাকেত পাের, তরাং তােদর লসা সেব া হেত পাের *lcm*(1*,*2*,...*10). এখােন লসা কীভােব আসল বুেঝছ তা? দখ, থম জন তার *a* + *b* সময় পর র অব ায় আসেব, ি তীয় জন তার *a* + *b* সময় পর তার র অব ায় আসেব এরকম কের সকেলর জ আমরা তার সাইেকেলর লংথ জািন। এখন যিদ করা হয় কত সময় পর সবাই একে র অব া আসেব তাহেল তা হেব এই সকল সাইেকল লংেথর lcm এর সমান। কারণ lcm ক সব সাইেকল লংথ ভাগ কের। তার মােন lcm সময় পর সবাই র অব ায় আসেব।

তাহেল তুিম যা করেত পােরা তাহেলা 1, 2 এভােব সই লসা পয িতিট সমেয় যেত পােরা এবং সবার ঘুেমর state দখেত পােরা। এভােব লসা পয সব সময় চক কেরও যিদ সবাই জেগ আেছ এরকম কান সময় না পাও তার মােন এমন কােনা সময় পাওয়া স ব নয়।

যারা আবার অলস তারা আবার ক কের lcm বর করেত যােব না। স জােন য কােনা একিট বড় মান পয চক করেলই হল। তরাং স চাখ ব কের 106 বা 107 জাতীয় বড় একিট সংখ া পয চক করেব। যিদ এর িভতেরও সমাধান না পাওয়া যায় তার মােন সমাধান নই!

কাডেক র করার জ চাইেল "অমুক সমেয়" "অমুক জেনর" state িক তা বর করার জ একিট ফাংশন িলখেত পােরা। অথ াৎ ঐ ফাংশনেক বত মান লােকর a, b এর মান আর বত মান সময় দেব, স বেল দেব য ঐ লাক ঘুিমেয় আেছ না জেগ। এভােব ফাংশন ফাংশন কের আলাদা আলাদা কাড করেল কাড বুঝেত বা িডবাগ করেত িবধা হয়।

একটা উদাহরন দই। িকছুিদন আেগ এক কে টে এক সম ায় আমােক বেলিছল য n ইনপুট আকাের দওয়া থাকেব। তামােক সই n এর উপর িভি কের একটা সম া সমাধান করেত হেব এবং একিট িল এর সাইজ এবং সই িল িট আউটপুট করেত হেব। কাড করার সময় আিম ভেবিছলাম সম াটা তা সহজ। তরাং ক কের র কের কাড না কের সব কাড main এর ভতেরই কির।িক যখন WA খলামতখন হলআসল ঝােমলা।িডবাগ এর জ আিমচাইিছলাম n এর িবিভ মােনর জ িলে র সাইজ ি ট দই। িক সম া হল যেহতু আিম র কের কাড কির নাই সেহতু দখা গেছ আিম িবিভ কস িবিভ ভােব া ডল কেরিছ। আমার আউটপুট ছিড়েয় িছিটেয় িবিভ জায়গায় আেছ। স ল জায়গা আবার সহেজ কেম ট করা যাে না। এরকম নানান সম া। শেষ আিম একিট ফাংশন িলখলাম এবং সই ফাংশন হেত vector িরটান করলাম। এবার আমার কাজ সহজ হেয় গল। আমার আউটপুট নানা জায়গায় আর ছিড়েয় িছিটেয় নাই। আিম যিদ েতই এই কাজ করতাম তাহেল আমার এেতা সময় ন হত না। তরাং চ া করবা তামার কাড থেকই র কের আলাদা আলাদা ফাংসেন ভে ভে িলখেত। আবার খয়াল রখ বিশ র করেত িগেয় যন অেনক বিশ সময় চেল না যায়! সবই trade-off!

UVa 253 Cube painting

**সমস া:** িট ঘনেকর িতিট পৃে র রং দওয়া আেছ। বলেত হেব এেদর একিট ঘনকেক ঘুিরেয় িফিরেয় অ ঘনেকর সােথ সদৃশ হয় এরকম orientation এ আনা যােব িকনা। অথ াৎ ঘুরানর পর িট ঘনেকর উপেরর িপেঠর রং একই হেব, িনেচর রং একই হেব, ডান বাম সামেন িপেছ সব পৃে িট ঘনেকর রং একই হেব- এরকম অব ােন আনা যােব িকনা।

**সমাধান:** এটা আমার বশ পছে র সম া। কারণ যিদ কউ "সহেজ" এটার কাড না করেত পাের তাহেল অ ত স অেনক খেট খুেট অেনক িবশাল কাড কের সমাধান করেত পারেব। আবার কউ একটু বুি খািটেয় কাডটােক একটু ছাট কের ফলেত পাের। কউ আবার একটু বিশ বুি খািটেয় আরও ছাট করেত পারেব। যত বিশ বুি খাটােব তত ছাট হেব। িক সই বুি িল বর করেতও

১৬

িক সময় লাগেব। তরাং trade-off! তুিম িক সময় ব য় কের কাড ছাট করেব, নািক dumb এর মত কম বুি ওয়ালা সমাধান িলখেব?

বড় কাড এবং সবেচেয় কম বুি ওয়ালা সমাধান কী তােতা বুঝেতই পারছ? িনেজ থেক কােনা ঘনকেক যত ভােব পােরা ঘুরাও আর তােদর সবার জ একিট কের if-else িলখ। অথ াৎ মেন মেন থম ঘনকেক যত ভােব স ব ঘুরাও। ঘুরানর পর তুিম জান এখন এর উপর/িনচ/ডান/বাম/সামেন- /িপেছ কান রঙ আেছ। এখন তুিম সই রঙ িলেক ি তীয় ঘনেকর রেঙর সােথ তুলনা করার জ if-else িলখ। যিদ ভুল না কের থািক তাহেল এজ তামােক 24িট if িলখেত হেব (আর তার ভতের 6িটর মত == লিজক)। ি তীয় ঘনকেক তা আর ঘুরানর েয়াজন নই। ধু থম ঘনকেক মেন মেন ঘুরােব। থম ঘনেকর ছয়িট পৃ েক এেক এেক মেন মেন সামেনর পৃ িহসােব ি র ক না করেব। এখন এর বািক পৃ িলেক ঘুিরেয় ঘুিরেয় পিরবত ন কের িবিভ orientation পেত পােরা। যিদ বুঝেত সম া হয় তামরা xyz axis ক না কর। ঘনেকর ছয়িট পৃ ছয় axis এ আেছ (x+, x-, ...). এখন মেন কর x+ এর িদেকর পৃ ি র। তাহেল এই ঘনকেক x axis এর সােপে ঘুরােল মাট চারিট orientation পাওয়া যােব (চারিট পৃ এেক এেক y+ এর িদেক আসেব- এভােবও ভাবেত পােরা)। িনঃসে েহ খুব একটা সহজ কাজ না। এেক তা ঘনকেক ঘুিরেয় ঘুিরেয় আঁকেত হেব কাগেজ কলেম বা ক নায়, তারপর আবার সটা এক গাদা if else কের িলখেতও হেব। তেব আর কােনা বুি না পেল এটাই করেত হেব। একটু র বা সহজ করার জ যা করেত পােরা তাহেলা এত েলা if-else না িলেখ একিট 24 *×* 6 সাইেজর অ ােরেত সংখ া িল িলেখ রােখা (অথ াৎ ি তীয় ঘনেকর i তম পৃে র সােথ ক িমেল এই তথ িল)। তাহেল একিট loop চািলেয় এবং একিট if িলেখই কাজ

সের ফলেত পােরা।

একটু কম ugly উপায় হেলা খয়াল করা য ঘনেকর পৃ িল একটু িবেশষ ভােব না ািরং করা (statement এর িচ দেখা)। 1 আর 6 িবপরীত িদেক, 2 আর 5 িবপরীত িদেক এবং 3 আর 4 ও। এেদর েত ক জাড়ার যাগফল িক 7. তরাং, আমরা যিদ িঠক কির উপের, িপছেন আর ডােন ক তাহেল আমরা সখান থেক িনচ, সামেন আর বােম ক আেছ তা বর কের ফলেত পাির। তরাং এর আেগর মত 24 *×* 6 এর দরকার নই 24 *×* 3 ই যেথ , কারণ বািক িতনিট তুিম এই ফমু লা ব বহার কের বর করেত পারেব।

আর কীভােব সহজ করা যায়? খয়াল কর, তুিম যিদ উপেরর পৃ িঠক কের ফলতাহেল চািরিদেক িক িক থাকেব তা িক fix হেয় গেছ। যমন উপের যিদ 1 থােক তাহেল চািরিদেক থাকেব 2, 3, 5, 4. এখন এেদরেক যিদ circular ভােব একিট অ াের ত রােখা তাহেল িক পাশাপািশ িট িসেল কের তােদরেক িপছন ও ডান পৃ ক না করেত পােরা। অথ াৎ িতিট পৃ উপেরর িদেকর জ তুিম একিট অ াের (4 দেঘ র) রােখা য সে ে চািরদেক কান কান পৃ থাকেব। যমন উপের 1 থাকেল চািরিদেক থাকেব 2, 3, 5 আর 4. উপের 4 থাকেল চািরিদেক থাকেব 2, 1, 5 আর 6 এরকম কের সকল পৃ উপের থাকার জ চািরিদেক ক থাকেব তার একিট িল বািনেয় ফলেত পাির। এরপর এই অ ােরর উপর লুপ চািলেয় িনব াচন করেত পাির িপছেন ক থাকেব, তাহেল তার পেরর জন থাকেব ডােন। আর উপর, িপছন ও ডান িনিদ হেয় গেল তা বািক েলা িনিদ হেয় যায় তাই না? তাহেল এই প িতেত কত বড় অ াের িলখেত হে ? 6*×*4. বশ কম! হয়েতা এর থেকও কমােনা স ব। িচ া কের দখেত পােরা, অথবা এটাই কাড কের ফলেত পােরা! সময় বাঁচেব।

Timus 1005 Stone Pile

**সমস া:** n িট সংখ া দয়া আেছ যখােন *n ≤* 20 এবং সংখ া িল সেব া 100*,*000 হেত পাের। তামােক এই সংখ া িলেক ভােগ ভাগ করেত হেব। এবার িতিট ভােগর সংখ া িলেক যাগ কর, ধরা যাক যাগফল িল হেলা A ও B. তামােদর ল হেব A ও B এর পাথ ক যন সবেচেয় কম হয়।

**সমাধান:** অেনকভােবই এই সম া সমাধান করা যায়। খয়ালকর সংখ া িল িক বশ ছাট(100*,*000)। তাই আমরা এেক coin change dp এর মত কের সমাধান করেত পাির। দ সংখ া িল খুব জাড় একবার ব বহার কের আমরা কান কান যাগফল বর করেত পাির - এই সম ােক dp এর মাধ েম

১৭

কের ফলা যায়।১ এই DP করেল আমরা জানেত পারব A এবং B এর িক িক মান স ব। আর এ িজিনস জানেল তা আমােদর সমাধান হেয়ই যােব!

এছাড়াও আমরা চাইেল এেক backtrack কের সমাধান করেত পাির। আমরা এেক এেক 1 হেত n পয িতিট ােন িগেয় িস া িনব য এই ােনর সংখ া কান ভােগ নব। ধরা যাক আমােদর backtrack ফাংশেনর parameter হেব িট সংখ া। থমিট বলেব আমরা এখন কান সংখ ায় আিছ (আমরা িক এেক এেক 1-তম হেত n-তম সংখ ায় যাি ) আর ি তীয় সংখ া বলেব আমরা যই ভােগ ভাগ করিছ তার থম ভােগ থাকা সংখ া িলর যাগফল কত। চাইেল ি তীয় ভােগর যাগফেলর জ আরও একিট প ারািমটার িনেত পার আবার চাইেল সবেশেষ িগেয় থম ভােগর যাগফলেক মাট যাগফল হেত বাদ িদেলই তুিম ি তীয় ভােগর যাগফল পেয় যােব। যাই হাক, এভােব করেল আমােদর time complexity O(2*n*) হেব। কারণ আমরা n িট সংখ ায় িগেয় ভােব চ া করিছ। থমবার চ া করিছ এেক থম ভােগ ফলেত, পেরর বার চ া করিছ এেক ি তীয় ভােগ ফলেত। তরাং n বার আমরা 2 ভােব চ া করিছ অথ াৎ 2*n* (যিদ " কন 2*n*"- এটা বুঝেত সম া হয় তাহেল তামরা combinatorics বা discrete math িনেয় একটু পড়া না করেত পােরা)।

Backtrack না কেরও আমরা খুব সহেজ এটা সমাধান করেত পাির। 0 হেত 2*n −* 1 এর মােঝ িতিট সংখ ার binary representation ক না কর২। অথ াৎ এই সংখ া িলেক n bit এর binary সংখ া িহেসেব ক না কর। n িট 0 আর 1 িদেয় তির সব সংখ া তুিম পেয় যােব। আমরা এেক এেক এই সকল সংখ া িলেক েসস করব। মেন কর একিট সংখ া িনলাম। এবার এর 0 ওয়ালা ান িলেক থম েপ আর 1 ওয়ালা ান িলেক ি তীয় েপ ফল। িত ভােগর যাগফল বর কর। এরপর তােদর মােঝর পাথ ক বর কর। উদাহরণ প মেন কর য আমােদর n হেলা 2. তাহেল 0 হেত 22*−*1=3 পয সংখ া িল (এবং তােদর binary প) হেলা 0(00)*,*1(01)*,*2(10)*,*3(11). 00 এর মােন হেলা আমােদর িট সংখ ােকই থম েপ িনেয়িছ, 10 এর মােন হেলা আমরা থম সংখ ােক (আশা কির একটু িচ া করেল বুঝেত পারেব য i তম least significant bit, i তম সংখ া িনেদ শ কের যখােন আমরা সংখ া িলেক 0 indexed ভােব ক না করিছ, মােন থম সংখ া আেছ 0 তম িবেট) থম েপ আর ি তীয় সংখ ােক ি তীয় েপ ক না করিছ। এভােব বািক 2 আর 3 কউ একই ভােব েসস করেত পাির। এই সমাধােনর time complexity হেব O(*n*2*n*) কারণ তামরা 0 হেত 2*n −* 1 এর লুপ চালােনার পর আরও একিট n দেঘ র লুপ চািলেয় 0-1 দেখ দেখ ভােগ ভাগ করছ। Backtrack এর ে ও তুিম যিদ ভাগ করার পর আবার লুপ চািলেয় দখেত কােক কান েপ ফেলছ তাহেলও এই অিতির n চেল আসেতা তামার complexity ত। িক উপেরর backtrack সমাধােন আমরা তা না কের যখন কােনা সংখ ােক থম ভােগ ফেলিছ তখনই থম ভােগর যাগফলেক আপেডট কেরিছ। ফেল আমােদর আর অিতির n এর লুপ লােগ নাই। িক আমােদর একটু আেগ বলা bitmask এর সমাধােন আমরা অিতির একিট লুপ চালাি 0-1 এ ভাগ করার জ । এজ আমােদর সমাধােন অিতির n চেল এেসেছ।

এখন কথা হেলা bitmask প িতর সমাধান হেত আমরা অিতির n বাদ িদেত পাির িক না। মেন কর mask নােমর ভ ািরেয়বলেক (bitmask এর সংি প) 0 হেত 2*n−*1 পয লুপ চালাি এবং িতিট mask এর জ আমরা ই ভােগর সংখ ার যাগফল বর কির (আমরা mask এর িবট িল দেখ িঠক কের ক কান ভােগ যােব)। এর আেগই বেলিছ, আমােদর আসেল ই ভােগর যাগফল রাখার দরকার নই, ধু এক ভােগর যাগফল জানেল আমরা মাট যাগফল হেত অপর ভােগর যাগফল সহেজই বর করেত পারব। তরাং মেন কর আমরা ধু যই ভােগ 1 িবট ওয়ালা সংখ া আেছ তােদর যাগফল জািন। এখন ধর আমরা এখন *mask* = *x* এ আিছ। এর মােন *mask < x* এর জ ইেতামেধ ই আমরা 1 িবেটর ভােগর সংখ ার যাগফল জািন। এখন যখন *mask* = *x* এ এেল তখন এর যেকােনা একিট on (1) িবট নাও। এেক শূ কের দাও (*mask*^ = 1). এই নতুন mask এর sum িক তুিম জােনা কারণ on িবটেক শূ কের দয়ার ফেল তামার নতুন mask িট বত মান mask এর থেক ছাট হেয় িগেয়েছ। তাহেল এই ছাট mask এর জ আমােদর আেগ পাওয়া যাগফল এর সােথ আমােদর on িবেটর সংখ া যাগ কের আমরা নতুন mask এর জ

যাগফল বর কের ফলেত পাির। িক কথা হল এই on িবট বর করেতই তা আমােদর n এর

১ া ািমং কে ট : ডটা াকচার ও অ ালগিরদম বইেয়র Dynamic Programming সকশেন Coin change dp এর নানা variation পােব।

২এই প িতেক আমরা bitmask প িত নাম িদেত পাির কারণ আমরা "all possible" bitmask এর উপর িদেয় লুপ চালাি ।

১৮

একিট লুপ লাগেব। না তা লাগেব না যিদ তুিম একটু বুি করেত পােরা। খয়াল কর আমােদর িক

যেকােনা on িবট িনেলই চলেব। আমরা িক খুব সহেজ বলেত পাির অমুক সংখ ার অমুক িবটটা on আেছ? পাির, 2 হেত 3 এর 2nd bit টা সবসময় on, 4 হেত 7 এর 3rd bit টা, 8 হেত 15 এর 4th bit টা এরকম কের 2*i−*1 হেত 2*i −* 1 এর i তম িবটটা সব সময় on. তাহেল আমরা সরাসির 0 হেত 2*n −* 1 এর লুপ না চািলেয় 0 হেত *n −* 1 পয i এর একটা লুপ চালােবা। আর এর ভতের আেরকটা লুপ চালােবা সই সই mask িদেয় যােদর i তম িবট আমােদর এই observation এ on অথ াৎ 2*i* হেত 2(*i*+1) *−* 1 পয । তাহেলই complexity O(2*n*) হেয় যােব। তামােদর িবধার জ backtrack আর bitmask প িতর কাড ২.২ এ দওয়া হেলা। তামরা চাইেল এেক অ ভােবও কাড করেত পার।

িফিরি ২.২: timus1005.cpp ১ int num[22], n; ২ int onBit[1 *<<* 20], sum[1 *<<* 20]; ৩৪ void backtrack(int at, int group1Sum) *{* ৫ if (at == n) *{* ৬ // Find group2Sum ৭ // Keep a global variable to track answer. ৮ return; ৯ *}* ১০ // Don t take num[at]. ১১ backtrack(at + 1, group1Sum); ১২ // Take num[at]. ১৩ backtrack(at + 1, group1Sum + num[at]); ১৪ *}* ১৫১৬ int main() *{* ১৭ scanf("%d", &n); ১৮ for (int i = 0; i *<* n; i++) *{* ১৯ scanf("%d", &num[i]); ২০ *}* ২১২২ bktk(0, 0); // Solution in backtrack. ২৩২৪ // Solution in bitmask. ২৫ sum[0] = 0; ২৬ for (int i = 0; i *<* n; i++) *{* ২৭ for (int mask = (1 *<<* i); mask *<* (1 *<<* (i + 1))*←↪*

; mask++) *{* ২৮ // i is the on bit. ২৯ sum[mask] = sum[mask ˆ (1 *<<* i)] + num[i]; ৩০ // Update answer, one part is: sum[mask] *←↪*

and another part is ৩১ // total *−* sum[mask]. ৩২ *}* ৩৩ *}* ৩৪ return 0; ৩৫ *}*

১৯

শষ করার আেগ একটা িজিনস। তামরা চাইেল ধের িনেত পােরা য n তম িজিনস সবসময় থম েপ থাকেব, এরপর বািক িজিনস িলেক সব ভােব ভােগ ভাগ করেব। তাহেল complexity O(2*n−*1) হেয় যােব।

Timus 1014 Product of Digits

**সমস া:** 0 হেত 109 রে র একিট সংখ া N দয়া থাকেব। তামােক সবেচেয় ছাট ধনা ক সংখ া Q আউটপুট িদেত হেব যন Q এর অ িলর ণফল N হয়। যিদ এমন কান সংখ া না পাওয়া যায় তাহেল *−*1 ি ট করেত হেব। যমন N = 10 এর জ Q হেব 25.

**সমাধান:** থমত দখ Q এর িডিজট িল ন করেল N পাওয়া যায়। অথ াৎ আমােদর জানা দরকার কান কান সংখ া িদেয় N ক ভাগ করা যায়। যিদ N ক 2*,*3*,...*9 িদেয় যতবার ভাগ যায় ততবার ভাগ দবার পরও দেখা য N তখনও 1 হয় নাই এর মােন এর জ আসেল কােনা Q পাওয়া স ব নয়। যমন N যিদ 26 হয় তাহেল দখেব 2 হেত 9 সব সংখ ািদেয় যত বার ভাগ দওয়া স ব ভাগ দওয়া হেয় গেল 13 বািক থােক। এর মােন এর জ আসেল কােনা Q খুঁেজ পাওয়া যােব না। কন? কারণ Q এর িডিজট িলেতা আসেল 2 হেত 9 ই হেব তাই না? 0 আর 1 িবেশষ ছাড়া ব বহার য হেব না তা মেন হয় একটু িচ া করেলই বুঝা যায়। তরাং কােডর েতই তােদর "specially handle" কের ফল। তাহেল আমােদর সংখ া ধুমা 2 হেত 9 িদেয় ভাগ যােব। বা আমরা আেরকটু িচ া কের বেল িদেত পাির য N সংখ ািট 2, 3, 5 আর 7 prime িল িদেয় ভাগ যােব, আর অ কােনা prime িদেয় নয়। তরাং আমরা বর কের ফিল N ক 2, 3, 5 আর 7 িদেয় কয়বার কের ভাগ করা যায়। এখন খয়াল কর আমরা চাইেল 3িট 2 ক এক কের 8 বানােত পাির, এেত কের আমােদর উ র এর িডিজট সংখ া িক 2 ঘর কেম যায়। আবার আমরা চাইেল িট 2 ক এক কের 4 বানােত পাির, 2 আর 3 ক এক কের 6 বানােত পাির অথবা িট 3 ক এক কের 9 বানােত পাির। তরাং তামােক িচ া করেত হেব তুিম কীভােব এেদরেক combine করেব যােত সবেচেয় ছাট সংখ া পাওয়া যায়। আর আশা কির এটা বুঝছ য িডিজট িলেক ছাট হেত বড় অড াের সািজেয় Q বানােত হেব (758 বানােনার চেয় 578 বানােনা লাভজনক)। এই পয ােয় এেস তুিম িনেজ থেক একটু িচ া কর কীভােব তুিম 2, 3, 5 ও 7 দর combine করেব। আসেল 5 আর 7 ক তা combine করার িকছু নই, combine করেত হেব 2 আর 3 ক। কারণ 5 আর 7 ক কারও সােথ combine করেত গেল তা আর 1 িডিজেট থােক না।

এখন নানা case িচ া করা ছাড়া আসেল কােনা উপায় নই। যিদ ধু একটা 2 বা একটা 3 থােক তাহেল তা িকছু করার নই। যিদ ধু এক গাদা 2 থােক তাহেল তা িতনটা িতনটা কের িনেয় যত ণ স ব 8 বানােনা লাভজনক। যিদ এরপর 4 বানােনা স ব হয় তাহেল 4 নাহেল আবার 2.

কন? তুিম 21*,*22*,*23 *...*210 এরকম 10 িট N এর জ িচ া কর তাহেলই বুঝেব। কে টে সবসময় মাণ করার দরকার হয় না। যিদও এটা িঠক য মাণ না কের কােনা িকছু করা risky িক তাও, মাণ অেনক সময় সােপ ব াপার। এর থেক িকছু উদাহরন হােত হােত কের যিদ তুিম িনেজেক বুঝ িদেত পার য এটাই িঠক- তাহেলই হেব। এর মােন এই না য এরকম করেল ভুল হয় না! যাই হাক, একই ভােব 3 এর ে ও িচ া করেত পার তেব সটা অেনক সহজ। এখন সম া হেব যখন 2 ও থাকেব আবার 3 ও থাকেব। আবােরা িকছু ণ case analysis করেল বুঝেব থেম 9 বানােনার চ া করেত হেব, এরপর 8, এরপর 6, 4 এরকম। আেগই বেলিছ কন এরকম হেলা, মাণ িক, এরকম ে কীভােব িচ া করব এতসব কের সবসময় লাভ নই। Case analysis কের intuition এর িভি েত সমাধান দাঁড় করােত হেব।

যিদ codejam বা hackercup এর মত contest হয় যখােন একবারই submit করা যায় সসব ে তামরা চাইেল একিট dp বা backtrack কাড কের দখেব তামার claim টা িক ধরা যাক N = 1 হেত 1000 পয সিঠক িকনা। যিদ হয় তাহেল তুিম মাটােমািট সাহস পােব য হয়েতা

তামার সমাধান বড় N এর জ ও সিঠক।

২০

Timus 1020 Rope

**সমস া:** N (*≤* 100) িট একই মােপর খুঁিটেক (সব খুঁিটই R ব াসােধ র বৃ ) একিট convex polygon এর N িট িব েত রাখা আেছ। একিট দিড়েক এই খুঁিট িলর চারপাশ িদেয় টানটান কের টানােনা হেল

সই দিড়র দঘ কত? **সমাধান:** খুবই চমৎকার একিট সম া। খয়াল কর দিড়র িকছু অংশ টানটান আর িকছু অংশ গাল গাল। টানটান অংশ হল িট বৃে র মােঝ শ ক। যেহতু বৃ িল একই ব াসােধ র সেহতু শ েকর দঘ তােদর কে র দূরে র সমান হেব। এর মােন দিড়র সাজা অংশ িলর দঘ হেব convex polygonএর পিরিধর সমান। িক গালঅংশ িল? এটাও খুব একটা কিঠন না। খয়ালকর সব েলা গাল অংশ জাড়া লাগােল একিট বৃ হয়। এটােক এভােবও িচ া করেত পােরা, মেন কর তুিম দিড় ধের হাঁটা করেল। গাল অংশ েলােত এেস তুিম R ব াসােধ র বৃ ধের একটু ঘুরছ এভােব করেত করেত আবার র জায়গায় আসেব। এই পেথ তুিম িক ঘুের ঘুের 360*◦* ঘুেরছ বা R ব াসােধ র বৃ পুেরাটা ঘুেরছ। বা তুিম চাইেল মেন মেন ল া অংশ িল বাদ িদেয় গাল অংশ িলেক জাড়া লাগােয় দখেত পােরা। তরাং আমরা যিদ বৃে র পিরিধ এবং পিলগেনর পিরিধ যাগ কির তাহেল মত দিড়র দঘ পেয় যাব।

Timus 1044 Lucky Tickets. Easy!

**সমস া:** কত িল n (*≤* 9 এবং n একিট জাড় সংখ া) িডিজেটর সংখ া আেছ যােদর অ িলর মােঝ থম অেধ েকর যাগফল শষ অেধ েকর সমান হয়। যমন 1322 এখােন 4 িট অংক আেছ যার থম িট অংেকর যাগফল আর শষ িট অংেকর যাগফল সমান।

**সমাধান:** এসব সম ার ে সবার আেগ দেখ িনেব য leading zero িক allowed িকনা। যমন এই সম ার ে allowed. এখন n যিদ জাড় হয় তাহেল সম াটা বশ সহজ। *n*/2 দেঘ র সব িল সংখ ার জ তার অংক িলর যাগফল আমােদর বর করেত হেব। এরপর আমরা বর করব কান যাগফল কতভােব হয়। ধরা যাক S যাগফল হয় k ভােব। তাহেল k ভােব থম অেধ ক আর আরও k ভােব পেরর অেধ েক আমরা S যাগফল ওয়ালা সংখ া বসােত পাির, অথ াৎ *k*2 ভােব আমােদর ই অেধ েকর অংেকর যাগফল একই হেত পাের। এভােব িতিট S এর জ আমােদর *k*2 বর কের তােদর যাগ করেলই হেয় যােব। আর কীভােব *n*/2 দেঘ র সকল সংখ া িল েসস করেব? তােতা বশ সহজ, Timus 1005 এর সমাধােনর মত তামরা চাইেল backtrack করেত পােরা, বা চাইেল লুপ চািলেয়ও করেত পােরা। যেহতু n এর মান সেব া 9 সেহতু *n*/2 এর সেব া মান মা 4. অথ াৎ চার িডিজেটর সকল সংখ ার যাগফল আমােদর বর করেত হেব।এটা তা কােনা ব াপারই না তাই না? মেন কর তামরা লুেপর মাধ েম *n*/2 দেঘ র সকল সংখ ার অংক িলর যাগফল বর করছ। অথ াৎ 0 হেত 10*n*/2*−*1 পয লুপ। আমরা িক চাইেল Timus 1005 এর মত এই লুপ চালােনার পর অংক িলর উপর লুপ নাও চালােত পাির। মেন কর আমরা চাইিছ x এর অংক সমূেহর যাগফল বর করেত। আমরা িলখেত পাির *sum*[*x*] = *sum*[*x*/10] + (*x*%10). তাহেলই িক x এর অংক িলর যাগফল বর হেয় যােব। আলাদা কের অংক িলর উপর লুপ চালােনার দরকার নই। এই সম ায় হয়েতা এই optimization এর দরকার নই, িক অ কােনা সম ায় কােজ লেগ যেত পাের।

Timus 1197 Lonesome Knight

**সমস া:** একিট দাবার বােড একিট ঘাড়ার ান দওয়া আেছ। বলেত হেব দাবার বােড র কয়িট ঘরেক এই ঘাড়া আ মণ কের আেছ। ইনপুট দওয়া হেব a1, g6 এরকম কের। **সমাধান:** থমত আমােদর ইনপুটেক েসস কের বর করেত হেব য আমােদর ঘাড়া কান row আর কান column এ আেছ। এই েসস করা তা বশ সহজ। তামরা ঘাড়ার ানেক একিট string

২১

এ ইনপুট নেব আর এই string এর থম character থেক a এর ascii মান বাদ দেব আর ি তীয় character হেত 1 এর ascii মান বাদ দেব (word[0] - 'a' এরকম)। মেন করা যাক আমােদর ঘাড়া আেছ (*r, c*) ত। এখন এখান হেত ঘাড়ািট কান কান ঘর আ মণ করেত পাের? এসব ঘরেক আমরা এভােব িলখেত পাির (*r* + 2*,c* + 1), (*r* + 2*,c −* 1), (*r −* 2*,c* + 1) এরকম কের মাট 8 িট। (*r* + 2*,c* + 1) এর মােন তা বুঝছ? এর মােন হল বত মান রা হেত ই রা িনেচ এবং বত মান কলাম হেত এক কলাম ডােনর ঘর। এরকম কের একিট ঘাড়া 8 িট মুভ িদেত পাের আর আমরা এই িতিট মুভ কই এরকম (*r* + 2*,c* + 1) এর মত কের কাশ করেত পাির।

িক সম া হল এই 8 িট ঘেরর িতিটই আমােদর দাবার বােড নাও থাকেত পাের। যমন আমােদর বত মান সল যিদ হয় a1 তাহেল িক (*r −* 2*,c −* 1) বেল িকছু নই। তরাং আমােদর চক কের দখেত হেব কান কান ঘর দাবার বােড আেছ (0 *≤ newR ≤* 7 এবং কলােমর ে ও একই ধরেনর চক)।

আমরা চাইেল এই নতুন অব ান িল হােত হােত িলখেত পাির। অথবা চাইেল সহেজ কাড করার জ row বরাবর অব ান পিরবত ন এবং column বরাবর অব ােনর পিরবত নেক িটঅ ােরেত রেখ িদেত পাির: *dr*[] = *{*2*,*2*,−*2*,−*2*,*1*,*1*,−*1*,−*1*}* ও *dc*[] = *{*1*,−*1*,*1*,−*1*,*2*,−*2*,*2*,−*2*}*. এরপর i এর একিট লুপ চালাব 0 হেত 7 পয এবং চক কের দখব (*r*+*dr*[*i*]*,c*+*dc*[*i*]) আমােদর

বােড র ভতের আেছ িকনা। এভােব খুব চমৎকার ভােব অেনক ছাট কের কাড হেয় যােব।

UVa 202 Repeating Decimals

**সমস া:** a ও b িট সংখ া দয়া আেছ (0 *≤ a ≤* 3000*,*1 *≤ b ≤* 3000). তামােক *a*/*b* ক

পৗনপিনক আকাের কাশ করেত হেব। যমন: 2/15 = 0*.*1(3).

**সমাধান:** এিট আমার অ তম পছে র একিট সম া। কারণ আমােদর দেশর বিশর ভাগ কেলজ বা িব িবদ ালয় পড়ুয়া ছা ছা ীেদর এিট সমাধান করেত িদেল বশ confusion এ পের যােব িক এর সমাধান ুল পড়ুয়ারাও বুঝেব বা চ া করেল িনেজ থেক বর করেত পারেব বা আসেল ুল কেলজ থেকই আমরা এই ব াপাের ধারনা িনেয় এেসিছ।

আমরা আসেল ভােগর সময় কী কির? ভাগ দই, ভাগেশেষর িপেছ একিট শূ বসাই (হর লেবর থেক বড় ধের িনি ), আবার ভাগ দই, আবার ভাগেশেষর িপেছ শূ বসাই এভােব চলেত থােক। এখন খয়াল কর যিদ কখনও ভাগেশষ পুনরাবৃি হয় তাহেল িক ভাগফেল আমরা যই িডিজট িল বিসেয়িছ তারাও িক পুনরাবৃি হেব, অথ াৎ পৗেনাপুিনক। এখন বলত যেকােনা a ক b িদেয় ভাগ করেলই কী এই পৗেনাপুিনক পাওয়া যােব? াঁ। কারণ তামার ভাগেশষ তা আসেল 0 হেত *b−*1 এর মােঝই হেব তাই না? যেহতু মা b রকেমর ভাগেশষ হেত পাের সেহতু এক সময় না এক সময় পুন াি ি হেবই। িক িব াস হে না? মেন কর আমরা 2 ক 15 িদেয় ভাগ করিছ। থেম তা ভাগেশষ 2 িনেজই। এর পর 5, এর পর 5 এবং 5 চলেতই থােক। এ জ আসেল ি তীয় িডিজেটর পর পৗেনাপুিনক চলেত থাকেব 0*.*1(3). এবার একটু কিঠন উদাহরণ নয়া যাক, 655/990. থেম ভাগেশষ 655, এর পর 610, এর পর 160, এর পর 610. তাহেল এর পর থেক 160, 610, 160, 610 এরকম চলেত থাকেব। অথ াৎ আমােদর থম িডিজেটর পর পেরর িট িডিজট বার বার পুনরাবৃি হেত থাকেব। আমােদর উ র হেব 0*.*6(61). যেহতু b খুব একটা বড় না সেহতু নানা ভােব এর সমাধান করা স ব।

মেন কর আমরা একিট 0/1 এর অ াের রেখিছ। েত সব ােন 0 থাকেব। এখন আমরা ভাগ দওয়া করব। ভােগর পের যই ভাগেশষ থাকেব আমরা অ ােরর সই ভাগেশেষর ােন িগেয় 1 িলেখ রাখব। িক যিদ দিখ আেগই 1 আেছ এর মােন আমােদর ভাগ সাইেকেল পেরেছ। আর আমরা এখন জেন গিছ এই সাইেকল কান ভাগেশষ থেক হেয়েছ, ধরা যাক এই ভাগেশষ হল x। এখন আবার আমরা এই কাজটু করব। আবার থেক ভাগ করা কির। যখন আমরা x ক ভাগেশষ িহসােব পাব এর মােন আমরা এখন সাইেকেল ঢুকিছ। এরপর আবার যখন x পাব, এর মােন আমােদর সাইেকল শষ। এভােব আমরা পুেরা ভাগফলেক পৗনপিনক িহসােব ি ট করেত পাির।

২২

২.১ অ শীলনী

২.১.১ সম া

SimpleX Timus 1264 Workdays X Timus 1293 Eniya

X UVa 1583 Digit Generator X Timus 1409 Two Gangsters X Timus 2012 About Grisha N X Timus 2023 Donald is a postman X UVa 10082 WERTYU X UVa 401 Palindromes X UVa 1584 Circular Sequence X UVa 1585 Score X UVa 1586 Molar mass X UVa 1225 Digit Counting X UVa 455 Periodic Strings X UVa 227 Puzzle X UVa 133 The Dole Queue X UVa 1368 DNA Consensus String X UVa 10340 All in All X UVa 1587 Box X UVa 1588 Kickdown X UVa 11809 Floating-Point Numbers X UVa 1339 Ancient Cipher X UVa 489 Hangman Judge X UVa 1591 Data Mining X UVa 815 Flooded! X UVa 272 TEX Quotes X Timus 1068 Sum X Timus 1082 Gaby Ivanushka

EasyX Timus 1025 Democracy in Danger X Timus 1313 Some Words about Sport

X Timus 1079 Maximum X Timus 1149 Sinus Dances X Timus 1319 Hotel X UVa 340 Master-Mind Hints X UVa 232 Crossword Answers X UVa 213 Message Decoding X UVa 512 Spreadsheet Tracking X UVa 201 Squares X UVa 220 Othello X UVa 1590 IP Networks X UVa 508 Morse Mismatches X UVa 509 RAID!

Medium

X UVa 12412 A Typical Homework X UVa 1589 Xiangqi

২.১.২ িহ ট

**UVa 272:** সম াটা কিঠন না। িবিভ জন িবিভ ভােব ইনপুট নেব। আিম য ভােব নই তাহেলা gets ব বহার কের। আিম আসেল scanf, printf আর gets এর বাইের িকছু ব বহার কির না ইনপুট আউটপুেটর জ । এসব ব বহােরর জ য সাবধানতা অবল ন করেত হয় তা িনেয় আিম ইেতামেধ ই " া ািমং কে ট ডটা াকচার ও অ ালগিরদম" বইেয় বেলিছ। তরাং তা আবােরা পুনরাবৃি করিছ না, ধু জািনেয় রাখলাম য gets ব বহার করেল একটু সাবধান হেত হয়।

**Timus 1068:** ভুেল যও না য N িক 1 এর থেক ছাট হেত পাের। পারত পে যেকােনা সম া সমাধােনর সময় ইনপুেট সবেচেয় ছাট মান এবং বড় মান িদেয় ট কের দখা উিচত।

**Timus 1025:** সংখ া িলেক সট কের িনেয় িচ া কের দখ।

**Timus 1313:** মেন কর তামার row িল উপর হেত িনেচ আর column িল বাম হেত ডােন 0 হেত *N −*1 পয না ার করা। তাহেল িনেচর বাম কাণা হেত উপেরর ডান কাণা পয য diagonal

২৩

যায় তােত থাকা ঘর িলর জ row এর id আর column এর id এর যাগফল িক একই হেব। আশা কির এই ি কটা ইেতামেধ ই তামরা জােনা। তাহেল একটু িচ া কের দেখা এই ি ক খািটেয় কীভােব সহেজ এই সম া সমাধান করা যায়। িট নে ড লুপ চািলেয়ই িক সম া সমাধান কের

ফলেত পারেব!

**Timus 1079:** n এর সবেচেয় বড় মান িদেয় ট করেত ভুেল যও না। ক জােন overflow হয় িকনা!

**Timus 1149:** ফাংশন, লুপ ইত ািদর ব বহার ঝালাই কের নবার জ খুবই ভাল একিট সম া। এখােন আসেল িশখােনার তমন িকছু নই। তুিম িনেজ একটু ধীের ে ঠা া মাথায় িচ া কের কাড কের ফল তাহেলই হেব।

২৪

অধ ায় ৩

STL াকিটস

STL এর পূণ প হেলা Standard Template Library. ব ল ব ব ত িকছু অ ালগিরদম (sort, binary search ইত ািদ) এবং ব ল ব ব ত িকছু data structure (queue, stack ইত া- িদ) এই STL এ আেছ। STL না জানেলও তুিম এই অধ ােয়র সম া িল সমাধান করেত পারেব। িক যই কাজ STL এ হয়েতা এক ই লাইেন হেয় যােব স কাজ STL ছাড়া করেত হয়েতা ১৫-২০ লাইন কের লাগেব। তরাং যখােন STL ব বহার করা যায় সখােন STL ব বহার করা উিচত। তেব STL অ ভােব ব বহার করা উিচত না। এর মােন এই না য তামােক জানেত হেব STL এর ভতের কীভােব কাড করা আেছ। যিদ এমন কােনা সময় আেস য তামােক STL ছাড়া সম া সমাধান করেত হেব তাহেল যন করেত পােরা। এই ান তামােক interview এর সময় বা কে টে র সময় দখা যােব িবিভ ভােব সাহায করেছ। আর মােঝ মােঝ STL েলা কীভােব implement করা আেছ তা জানেল ভাল হয়। তাহেল STL এর িবিভ complexity স েক ভাল ধারনা পাওয়া যায়। তামােদর যােদর আ হ আেছ তারা STL স েক আরও গভীের পড়েত পােরা নেট খাঁজ কের। এই অধ ােয় মূলত আমরা িকছু সম া দখব যা STL িদেয় সহেজ সমাধান করা যায়। তামরা চাইেল থেম STL স েক অেনক িকছু পড়েত পােরা, সব STL িশেখ ফলেত পােরা। িক তােত কের সম া হেলা সসব মেন থাকেব না। তেব তুিম যিদ স ভ করেত করেত শখ তাহেল সই STL এর ব বহার মেন থােক অেনক িদন এবং পরবত ীেত যখন অ কােনা সম ায় দরকার পড়েব তুিম চট কের STL ব বহার করেত পারেব। আর এ অধ ােয়র সম া িল আসেল কীভােব সমাধান করেত হেব সটা বলা মুল উে না। মুল উে হেলা কান সম ায় কান STL ব বহার করা যায় তা দখােনা এবং সই STL িনেয় িকছু আেলাচনা করা। তরাং তামরা চ া করেব এই অধ ােয়র িতিট সম ােক সখােন বলা STL ব বহার কের সমাধান করেত।

UVa 10474 Where is the Marble?

**সমস া:** তামােক N িট (*N ≤* 10*,*000) সংখ া দয়া হেব। সংখ া িল সেব া 10*,*000 হেব। থেম তামােক এই সব সংখ া ছাট হেত বড় সট করেত হেব। এরপর তামােক িকছু query দওয়া হেব ( ায় 10*,*000 িট query)। িতিট query ত একিট কের সংখ া q দওয়া হেব। তামােক বলেত হেব তামার সেট ড সংখ ার িলে q আেছ িক না, থাকেল থম কান ােন আেছ। যমন: 1, 3, 3, 5 এই সেট ড না ার িলেত 2 নই। আবার থম 3 আেছ 2nd ােন।

**সমাধান:** ধরা যাক সংখ া িল A নােমর একিট অ ােরেত ইনপুট নওয়া হেয়েছ। থেম আমরা এই সংখ া িলেক সট করব।এজ আমরা STL এর sort ফাংশন ব বহার করব। আর এর জ আমােদর algorithm হডার ফাইল include করেত হেব। সই সােথ "using namespace std;" এই কথািটও িলখেত হেব। সাধারণত সব হডার ফাইল include করা শেষ এই লাইনিট আিম িলেখ থািক। কন এই লাইনিট িলখেত হেব সটা আেরকটা িবশাল আেলাচনার জ িদেব। তামােদর আ হ থাকেল namespace িলেখ সাচ িদও। আর stl এর যেকােনা হডার ফাইেলর জ তামােদর এই

২৫

লাইন িলখেত হেব (একবার িলখেলই চলেব)। এখন কােড তুিম যিদ িলখ "sort(A, A + N)" তাহেল A[0] হেত A[N - 1] পয সট হেয় যােব। যিদ তামােদর সংখ া িল A[1] হেত A[N] থােক তাহেল তামােদর িলখেত হেব "sort(A + 1, A + N + 1)"। মাট কথা তামােদর যিদ A[a] হেত A[b] পয sort করেত হয় তাহেল িলখেত হেব "sort(A + a, A + b + 1)". কন এভােব িলখবা তা জানেত হেল pointer িনেয় একটু পড়া না করেত হেব তামােদর।

সট তা করা হেলা, িক কীভােব বর করেব য query এর সংখ া এই অ ােরেত আেছ িক না, থাকেল থেম কাথায় আেছ? এজ তামরা চাইেল িনেজ থেক binary search এর কাড করেত পােরা অথবা stl এর lowerbound বা upper bound ব বহার করেত পােরা। এেদর ব বহার করেতও algorithm হডার ফাইেলর দরকার। Lower bound আর upper bound এর কানটা য িক দয় তা আমার কখনই মেন থােক না। যখন ব বহার কির তখন নেট দেখ নই। Lower bound ও upper bound ই ে ই আমােদর একিট সট করা অ াের িদেত হয় আর একিট সংখ া িদেত হয়। Lower bound অ াের এর সই ান দয় যখােনর সংখ া query এর সংখ ার সমান বা বড় এবং এেদর মােঝ সবেচেয় আেগর ান দয়। আর upper bound দয় এমন ান যার সংখ া query এর সংখ া থেক বড় এবং এেদর মােঝ সবেচেয় আেগর ান দয়। যমন তামার অ াের যিদ হয় 1, 2, 2, 4, 5 তাহেল 2 এর lower bound যিদ call করা হয় সিট িদেব থম 2 এর ান, আর যিদ upper bound ক call করা হয় তাহেল িদেব 4 এর ান। এই lower bound বা upper bound এর ফাংশন কীভােব call করেত হয়? "lower bound(A, A + N, q)" যখােন A অ ােরেত 0 indexed ভােব তুিম স া িল রেখছ। একটা িজিনস, এইেয বললাম অমুেকর ান দেব - এই ােনর মােন িক? Index? না, ওই ােনর pointer. তরাং তামরা যিদ index চাও তাহেল এই lower bound এর return মান হেত A িবেয়াগ কের নেব। তাহেলই 0-indexed অ াের পিজশন পেয় যােব। অথ াৎ তামােক িলখেত হেব "lower bound(A, A + N, q) - A". তাহেল তুিম index পেয় যােব। িক যিদ আমােদর অ াের 0 indexed না হেয় 1 indexed হত তাহেল কীভােব কল করেত? িঠক sort ফাংশেন যভােব িশেখিছ সভােব "lower bound(A+1, A+N+1, q)".

এখন আশা যাক এেদর time complexity ত। অেনেক মেন কের এটা তা binary search এর মত, তরাং এর time complexity হেব লগািরদম। কথাটা িঠক আেছ আবার িঠক নই। সিত কথা বলেত সব সময় এর complexity লগািরদম না। অ াের বা ভ র বা এই জাতীয় random access করা স ব এরকম data structure এর ে এরা লগািরদম (random access করা যায় মােন constant সমেয় যেকােনা index ক access করা যায় এরকম)। অ ে এরা িলিনয়ার (O(*n*)) যমন list, set ইত ািদ।১ এসব ে linear সময় লােগ কারণ list বা set এসেবর ে i তম index এ ক আেছ এটা constant সমেয় বর করা স ব না। তাই এসব ে স এেক এেক সকল element দেখ এরপর lower bound বা upper bound এর মান িনণ য় কের।

যাই হাক, তামরা চাইেল আরও সহেজ এই সম া সমাধান করেত পারেত। সট করার পর তুিম একিট অ ােরেত িলেখ রাখেত পােরা য, কােনা একিট সংখ া সেট ড অ ােরেত কাথায় থম এেসেছ। এই কাজ তা িলিনয়ার সমেয়ই করা যায়। মেন কর সট করা অ ােরর শষ হেত েত যােব এবং িলখেব য place[A[i]] = i তাহেলই হেব। িক যিদ আমােদর ইনপুেটর সংখ া িল 10,000 এর থেক বড় হয়, অেনক বড়? িচ া কর- এে ে সম া িক? সম া হল আমরা এেতা বড় সংখ ােক index িহসােব ব বহার করেত পাির না বা করেত হেল যত বড় অ ােরর েয়াজন তত বড় অ াের আসেল নওয়া স ব না। এই অব া থেক মুি িদেত পাের stl এর map. মেন কর তামার কােছ িকছু নাম আেছ আর তার বয়স আেছ। তুিম M["hasan"] = 3 করেল hasan এর িবপে 3 থাকেব এই ম ােপ। তুিম যিদ পের কখনও (যতবার) M["hasan"] কর 3 পােব। ম াপ ব বহােরর জ তামােক map নােমর হডার ব বহার করেত হেব। এিট declare করার syntax হেলা "map<string, int> M;" অথ াৎ যই টাইপেক ম াপ করবা তা েত, আর যই টাইেপ ম াপ করবা তা শেষ। এখন তুিম একিট অ ােরর মত কের ব বহার করেত পােরা এেক। "M[?] = ?" কের তুিম map এ মান রাখেত পােরা আবার "? = M[?]" কের ম াপ থেক কােনা িকছুর মান

১Set এর ে এর িনজ lower bound ও upper bound আেছ। অথ াৎ যিদ "set<int> S" নােম একিট set তামার কােছ থােক তাহেল তামরা "S.lower bound(q);" এভােব lower bound ব বহার করেত পােরা এবং তা লগািরদিমক হেব। সাধারণ "lower bound(S.begin(), S.end(), q);" করেল িক িলিনয়ার সময় লাগেব।

২৬

বর করেত পােরা। িঠক অ ােরর মত। তরাং ইনপুেটর মান যিদ অেনক বড় হয় সে ে তামােক long long ক int এ ম াপ করেত হেব। Map এ কােনা িকছু রাখেত বা খুঁেজ বর করেত সময় লােগ O(log *n*) যখােন n হল map এর সই মুহূেত র সাইজ। আর n িট িজিনস রাখেত O(*n*) memory লােগ।

UVa 101 The Blocks Problem

**সমস া:** 0 হেত n - 1 পয n িট ক আেছ যখােন n সেব া 25 হেত পাের। এছাড়াও 0 হেত n - 1 পয n িট pile আেছ। েত িতিট ক তার id এর সমান id এর pile এ থােক। অথ াৎ i তম ক i তম pile এ থােক। এরপর এেক এেক িকছু instruction দওয়া হেব। Instruction িল 5 ধরেনর হেত পাের। "move a onto b" বলেল a কেক b েকর উপের িনেত হেব। এজ তােদর উপের যসব ক িছল তােদরেক তােদর র জায়গায় ফরত িদেত হেব। র জায়গায় ফরত িদেত হেব মােন তােদর র pile এ ফরত িদেত হেব। একটা িজিনস খয়াল কর এখােন িক বেল নায় য কীভােব ফরত িদেত হেব, সই জায়গায় িকছু থাকেল িক করেত হেব তা বলা নই। িচ া কের দখেত পার কন বলা নাই, না বলেল কােনা িত আেছ িকনা (আেগ পুেরা বেলমটা তা পড়!)। আেরকিট instruction "move a over b" বলেল a কেক b ক যই pile এ আেছ তার top এ রাখেত হেব। এসময় a েকর উপের যত ক িছল তােদরেক তােদর র জায়গায় ফরত িদেত হেব। "pile a onto b" বলেল b এর উপের থাকা সব কেক তােদর র জায়গায় ফরত িদেত হেব আর a ক আর তার উপের যত ক আেছ তােদরেক b েকর উপের এেন রাখেত হেব। "pile a over b" বলেল a ও তার উপেরর সব কেক b যই pile এ আেছ তার উপের এেন রাখেত হেব। "quit" বলেল আমােদর অপােরশন শষ কের কান পাইেল এখন কান কান ক আেছ তা িনচ হেত উপর এই েম ি ট করেত হেব।

**সমাধান:** Pile দেখই চট কের stack ব বহার করেত বেস যও না। কারণ আমােদর ধু pile এর উপেরর িজিনস না ভতেরর িজিনসও লাগেব। মেন কর a কােনা pile এর ভতের আেছ। তাহেল তা আমােদর ভতেরর িজিনস িলর উপর িদেয় লুপ চালােত হেব তাই না? াঁ, তামরা চাইেল অব যত ণ না a পাও তত ণ ঐ stack থেক তুেল আেরকটা কােনা জায়গায় রাখেত পােরা। তেব আিম িনেজ এই বেলেমর কাড করেল vector ব বহার করব। vector হেলা এমন একটা অ াের যা েয়াজন মত ল া হয়। এিট ব বহােরর জ তামােক আেগ থেক বেল িদেত হয় না য তামার কত সাইেজর vector লাগেব (চাইেল বলেত পার)। তামার যিদ নতুন কান সংখ া এর পছেন ঢুকােত হয় তাহেল বলেব "V.push back(10);" তাহেলই এর পছেন 10 ঢুেক যােব। আবার চাইেল "pop back" বেল পছন থেক সংখ া সরােতও পারেব। এসব িজিনস constant সমেয় হেব বেল ধের িনেত পােরা অথ াৎ কান সময় লাগেব না এসব করেত। এছাড়াও vector এর আেরকটা িবধা হেলা তুিম চাইেল "V[4]" বেল এর 4 index এর সংখ ােক access করেত পারেব। মাট কথা এিট একিট অ াের যা ব বহােরর েত তামােক সাইজ বেল িদেত হয় না। vector এর push back, pop back ছাড়াও আরও অেনক ফাংশন আেছ। এেদর মােঝ ব ল ব ব ত িল হেলা size (vector এর বত মান সাইজ িরটান করেব), clear (পুেরা vector ধুেয় মুেছ পির ার কের দেব) অ তম। সিত কথা বলেত এই চারিট ফাংশন ছাড়া আর িকছু আমার লােগ না। একিট কথা, vector এর জ erase, insert, find১ ইত ািদ িকছু ফাংশন আেছ। তামরা ভাবেত পােরা য "আহ কত র একিট অ ােরর মােঝ আমরা insert, erase করেত পাির বা vector এর ভতের find করেত পাির।" না ঐ কাজ করেত যও না। কারণ এেত insert বা erase বা find এর জ িলিনয়ার সময় লাগেত পাের worst case এ। কবল মা push back, pop back আর কােনা index এ *V* [*i*] এর মত কের access করা constant সমেয়র অপােরশন। আরও একিট কােজর ফাংশন আেছ আর তাহেলা resize. নাম েনই বুঝেত পারছ এ িক কের। 100 সাইেজর একিট vector V ক যিদ বল V.resize(50) তাহেল এর সাইজ 50 এ কেম যােব। আবার উলেটা ভােব যিদ 50 সাইেজর অ ােরেক বল V.resize(100) তাহেল আরও 50 িট ান যাগ হেয় যােব। নতুন জায়গা িলেত যতদূর স ব garbage value থাকেব। যিদ তুিম 0 িদেয় তােদরেক initialize করেত চাও তাহেল তামােক

১Find মেন হয় vector এর িনজ ফাংশন না। এিট algorithm হডার ফাইেলর একিট ফাংশন যা sort/lower bound ইত ািদর মত কের ব বহার করেত হয় অথ াৎ find(V.begin(), V.end(), x) এরকম।

২৭

িলখেত হেব V.resize(100, 0). তামােদর যিদ n সাইেজর একিট অ াের লােগ তাহেল n ইনপুট নবার পর V.resize(n) কের n সাইেজর অ াের বািনেয় ফলেত পােরা V ক। এর পর সাধারণ ভােব &V[i] বা V[i] িদেয় ইনপুট আউটপুট বা িহসাব করেত পােরা।

তাহেল তামরা এই সম া vector ব বহার কের সমাধান কের ফলেত পার। য কয়িট অপাের- শন দওয়া আেছ স েলা implement করেত হেল আমােদর জানেত হেব কান pile এ আমােদর সংখ া আেছ। সই কাজ তামরা িত pile এ িগেয় তােত লুপ চািলেয় কের ফলেত পারেব। আবার কােনা pile থেক যিদ উপেরর সংখ া সরােত চাও তাহেল pop back ফাংশন ব বহার করেব। কা- না pile এ নতুন সংখ া ঢুকােত হেল push back ফাংশন ব বহার করেত পারেব। এরকম কের vector ব বহার কের খুব সহেজই এই সম া সমাধান করা স ব।

UVa 10815 Andy's First Dictionary

**সমস া:** তামােক একিট ট ট দওয়া হেব। ট েট দািড় কমা বা আরও অ া punctuation সহ capital বা small letterএ িলখা অেনক শ থাকেব। তামােক এই ট েটথাকাসকল িভ িভ শ lexicographical অড াের সািজেয় ি টকরেত হেব। সই সােথ শ িলেক case insensitive িহ- সােব ক না করেত হেব। অথ াৎ apple, AppLE, APPLE সবই এক। ি ট করার সময় তামােক শ িলেক lower case এ ি ট করেত হেব।

**সমাধান:** যিদ তামার কােছ একিট শ থােক তুিম তা খুব সহেজই তােক lower case এ িনেত পারেব তাই না? যিদ মেন না থােক আবােরা বিল, িতিট character এর উপর লুপ চািলেয় তুিম দেখা য character িট upper case িক না ('A' <= word[i] && word[i] <= 'Z')। যিদ হয় তাহেল তােক lower case কর (word[i] = word[i] - 'A' + 'a'). এবার এই word ক stl এর set এ ঢুিকেয় দাও। আমােদর গিণেতর set এর মত stl এর set এও একই িজিনস একািধকবার থােক না। সই সােথ তুিম যিদ set এ লুপ চালাও তাহেল তুিম শ িলেক lexicographical এই সট করা পাবা। আমােদর একিট সট লাগেব যা string ক রাখেব। এর জ তামার েয়াজন set নােমর হডার ফাইল। এেক declare করেত হেব "set<string> S;" এই ভােব। খয়াল কর এখােন িক আমরা stl এর string ও ব বহার করিছ (string হডার ফাইল)। আিম ব ি গত ভােব কখনও char\* এর set ব বহার কির নাই তেব এটা বলেত পাির য char\* এর set ব বহার করেল আরও অেনক ঝােমলা করেত হেতা (িনেজ থেক comparison function িলেখ িদেত হেতা)। মেন কর তামার কােছ একিট শ আেছ "char word[20];" এ। তুিম এেক আমােদর set এ ঢুকােত চাও। এজ ধু "S.insert(word);" িলখেলই চলেব। এখােন char এর অ াের িনেজ থেকই string এ convert হেয় যােব। সব শেষ set এর সকল শ এর ভতর িদেয় লুপ চালােনার জ c++11 এর range based loop ব বহার করেলই চেল "for(string w : S)" এবার তুিম cout বা printf িদেয় w ক print কের িদেলই হেলা। cout িদেয় ি ট করা সহজ তেব এেত সময় বিশ লাগেত পাের। আিম এজ printf ব বহার কির। তেব stl এর string ক printf িদেয় ি ট করেত একটু ক করেত হয় printf("%s", w.c str());. মূল সম ায় ফরার আেগ set এর িকছু দরকাির ফাংশন বেল নই। size() - সট এর সাইজ এবং এে ে unique count দেব অথ াৎ তুিম যিদ একই িজিনস একািধকবার insert কের থােকা তা একািধকবার count হেব না, একবার count হেব। এছাড়াও empty() - সট িক এখন ফাঁকা? count(x) - x কয়বার আেছ, আসেল সেব া একবারই থাকা স ব তাই এিট হয় 0 বা 1 িরটান করেব। count এর complexity লগািরদিমক।

সবই তা বুঝলাম িক একটা ট ট থেক কীভােব শ িল পাব? এে ে িবিভ জেন িবিভ উপায় অবল ন কের থােক। আিম space ওয়ালা ট ট ইনপুেটর ে সবসময় gets ব বহার কের থািক। মেন কর একিট char line[100] অ ােরেত পুেরা লাইন ইনপুট িনেয়িছ। এরপর আিম সকল non alphabet character ক space এ পিরবত ন কের ফিল (একিট লুপ চালােলই হয়)। সই সােথ চাইেল upper case letter কও lower case এ পিরবত ন কের ফলা যায় (তাহেল আর পের upper case, lower case িনেয় ঝােমলা করেত হয় না)। এবার stl এর istringstream ব বহার কির (sstream হডার ফাইল)। এেক declare করেত হয় এভােব "istringstream iS(line);". এর মােন line অ াের হেত একিট istringstream (পূণ প input string stream) বানােনা হেলা, যার নাম iS. এবার আমরা এই লাইেনর িতিট শ েক আলাদা আলাদা কের চাই। খুবই সহজ-

২৮

while(iS >> w) যখােন w হেলা একিট stl এর string (string w). এবার এই শ েক stl এর set এ ঢুিকেয় িদেলই হেয় যােব।

set ও istringstream এর যা ব বহার দখালাম এর বািহের তমন কােনা ব বহার নই, অ তঃ আমার তমন লােগ না (set এর size ফাংশন ছাড়া)। set এ মােঝ মােঝ erase ব বহার করেত হয়, কােনা একিট িজিনসেক set হেত মুেছ ফলার জ । এছাড়া lower bound বা upper bound এর কথা তা আেগই বেলিছ। তুিম যিদ নেট কােনা একিট stl এর নাম িলেখ সাচ দাও তাহেল ব উদাহরন পাবা।

Timus 1209 1, 10, 100, 1000...

**সমস া:** একিট sequence দওয়া আেছ: 110100100010000*...*. বলেত হেব k তম অংক কত

যখােন 1 *≤ k ≤* 231 *−* 1.

**সমাধান:** যারা গিণত পছ কর তারা ফমু লা বর কের সমাধান করেত পােরা। থেম দখ কাথায় কাথায় 1 আেছ। 1, 2, 4, 7, 11 ইত ািদেত। চ া কের দখ এেদর কােনা ফমু লা আনেত পােরা িকনা। অেনক উপােয়ই ফমু লা বর করা যায়। যমন একটা উপায় হেত পাের pattern খয়াল কের: 1 = 1+(0), 2 = 1+(0+1), 4 = 1+(0+1+2), 7 = 1+(0+1+2+3). তরাং i তম সংখ া হেব 1+(0+1+ *...i*). ােকট এর ভতেরর অংেশর ফমু লা তা জােনাই! তরাং ইনপুট k এর জ আমােদর দখেত হেব এমন কােনা i আেছ িকনা। তা quadratic equation সমাধান করেলই সহেজই বর হেয় যােব।

ফমু লা বর না কেরও করা যায়। এটা খয়াল কর য এই sequence এ খুব বিশ 1 নই। পাশাপািশ িট 1 এর দূর linear হাের বাড়েছ। অথ াৎ থম িট 1 এর মােঝর র 0, ি তীয় িট 1 এর মােঝর র 1, এর পেরর র 2 এরকম কের 3, 4, 5... তাহেল 231 দেঘ র মােঝ আসেল *√*231 এর বিশ 1 থাকেব না। কীভােব বুঝলাম? সহজ, িকছু ণ আেগই বেলিছ আমােদর i তম 1 থাকেব *i*2 এ (একদম িঠক কের বলেল (*i*(*i* + 1)/2 + 1) এ)। তরাং আমরা বলেত পাির 231 এর ভতের মাট *√*231 িট 1 থাকেব। আমরা খুব সহেজই একিট লুপ চািলেয় সব 1 এর ান বর কের ফলেত পারব (মােন ফমু লায় না কের 0+1+*...i* এর কাজ লুেপ করা) এবং তােদর একিট set এ রাখেত পারব। এরপর িত ইনপুেট চক কের দখেত পােরা য set এ তামােদর ইনপুেটর সংখ া আেছ িকনা। চাইেল তামরা ক কের িনেজ থেক binary search ও করেত পােরা। অথবা stl এর count, find, lower bound বা upper bound ও ব বহার করেত পােরা।

UVa 156 Anagrams

**সমস া:** যিদ িট শে একই letter সমূহ একদম িভ অড াের থােক তাহেল তােদর anagram বেল। যমন SPOT আর POTS. এই সম ার ে অ র িল িভ অড াের এবং িভ case এ থাকেলও আমরা তােদর anagram বলব। আবার যিদ শ িটেত অ র িল একই অড াের থােক িক তােদর কােরা case যিদ আলাদা হয় আমরা তােকও anagram ধরব। যমন spot আর spot িক anagram না, িক spot আর SpoT আবার anagram. একিট ট েট যিদ কােনা একিট শে র anagram না থােক তাহেল তােক ananagram বেল। আমােদরেক একিট দ text এর সকল ananagram ক lexicographical অড াের ি ট করেত হেব।

**সমাধান:** এই সম ার বিশর ভাগ আেলাচনা UVa 10815 এর মত। ধু একিট িজিনস িনেয় কথা বলব। কীভােব বুঝেব য কােনা একিট শে র anagram এর আেগ এেসিছল িকনা। থমত িচ া কর িট শ anagram িকনা তা কীভােব বুঝেব। িট শ anagram হয় যিদ একই letter িল িভ অড াের থেক থােক বা একই অড াের থােক িক কােনা একিট অ েরর case আলাদা হয়। এর মােন শ িটেক একই হওয়া যােব না আর তােদর অ র িলেক sort করেল একই শ হেব (case insensitive comparison করেত হেব, অথ াৎ িট string compare এর সময় case

২৯

এর িভি েত িট অ রেক আলাদা মেন করা যােব না)। অথ াৎ S1 আর S2 একই হওয়া যােব না (S1 != S2) আর এেদেরক সট করার পর (sort(S1.begin(), S1.end()) ও S2 এর জ ও একই রকম) ও যিদ তােদর ভতের কােনা upper case letter থােক তােদর small case এ পিরনত করার পর S1 ও S2 একই হেব (S1 == S2). এইেয একিট শে র সকল character ক সট কেরছ এবং তার ভতের যিদ কােনা upper case letter থােক তােক lower case এ পিরনত করছ এেক বেল একিট শ েক canonical form এ আনা। এেত লাভ িক? লাভ হেলা- তামরা তা মুল শ দেখ সরাসির বুঝেতিছেল না য এই িট শ anagram িকনা িক তােদর canonical form দেখ সহেজই বুঝেত পারেব তারা anagram িকনা। অব এই সম ার ে canonical form এর পাশাপািশ তামােক জানেত হেব মুল শ িট িক। কারণ িট শ িনেজরাই যিদ একই হয় তাহেল তা আর তারা anagram হেব না।

এটু যিদ বুেঝ থােকা তাহেল আমার মেন হয় set ব বহার কের সম াটা সমাধান কের ফলেত পারেব। তাও বিল। িট শ anagram হওয়ার শত িট। এক- তােদর একই হওয়া যােব না, ই- তােদর ভতের একই অ র িল অ অড াের বা অ case এ থাকেত পাের। আমরা এেক এেক এই িট শত েয়াগ করব। থেমই আমরা আমােদর ইনপুেট দওয়া সব শ েক একিট সেট ঢুকাব। তা- হেল যিদ একই রকম িট শ থেক থােক আমরা তােদর পিরবেত একিট শ পাব। আমােদর থম শত পুরন হেয় িগেয়েছ। এবার ি তীয় শত । আমােদরেক িতিট শে র জ তার canonical form বানােত হেব।এখন এই একই canonical formওয়ালা শ িলেক আমােদর বাদ িদেত হেব।কীভা- ব? আমরা িট সট নব। একিট সেট থাকেব সসব canonical form যােদর anagram এখনও খুঁেজ পাওয়া যায় নায়। অথ াৎ ananagram শ িলর canonical form এর সট। আর আেরক সেট থাকেব anagram শ িলর canonical form. আমরা এেক এেক শ িল আর তােদর canonical form েসস করব। থেমই দখ নতুন শে র canonical form িক ananagram এর সেট আেছ িকনা। যিদ থােক এর মােন এিট anagram. আমােদরেক ananagram এর সট হেত সই canonical ফম েক মুেছ ফলেত হেব আর এই canonical form ক anagram এর সেট insert করেত হেব। িক যিদ এই শ ananagram এর সেট না থােক তাহেল আমরা দ- খব anagram এর সেট আেছ িকনা। যিদ থােক তার মােন এই শ ও anagram. যেহতু এই canonical form ইিতমেধ ই anagram এর সেট আেছ সেহতু আমােদর করার িকছু নই। িক anagram এর সেটও যিদ না থােক এর মােন এই শ িট আপাতত ananagram. তরাং এর canonical ফম েক আমরা ananagram এর সেট insert করব। এই কাজ যিদ আমরা সকল শে র জ কির তাহেল সব শ েসস শেষ আমরা সকল ananagram এর canonical form

পেয় যাব। এবার শষ কাজ- আমরা আবার আমােদর মূল শ আর তার canonical form এর িল ধের ধের যাব। যিদ দিখ তার canonical form িট ananagram এর সেট আেছ তার মােন এই শ িট ananagram. দাও ি ট কের। এরকম কের সকল ananagram ি ট করেত হেব। খ- য়াল রখ, আমােদর ি ট করা শ িল িক lexicographcal order এ ি ট করেত হেব। কীভােব করেব? যই শ িল ি ট করেব তােদর একিট set এ ঢুিকেয় তােদর ি ট করেত পার, বা একিট vector এ ঢুিকেয় তােদর সট কের এরপর ি ট করেত পার। অথবা এইেয শষ লুেপ আমরা যখন মুল শ আর তার anagram এর উপর লুপ চালাি লাম তখনই যিদ মূল শে র lexicographical অড াের লুপ চালাই তাহেল আর ক কের আেরকিট set বা vector িনেত হেব না। আসেল আমরা তা একিট set এ আমােদর সকল শ েক ঢুিকেয়িছলাম (আমােদর সমাধােনর থম সট) ঐ সেটর উপেরই সবসময় কাজ করেলই হেব, আমােদর আর অিতির set বা vector িনেত হেব না।

UVa 12096 The SetStack Computer

**সমস া:** মেন কর একিট সট এর stack আেছ। এেক এেক তামােক সেব া 2000 িট অপােরশন বলেব। Push বলেল তামােক একিট ফাকা সট *{}* পুশ করেত হেব। Dup বলেল তামােক stack এর উপেরর যই সটিট আেছ তা আবােরা পুশ করেত হেব। Union বা Intersectবলেল stack হেত উপেরর িট element ক pop কের তােদর union বা intersection ক পুশ করেত হেব। Add বলেল একই ভােব stack এর উপেরর িট element িনেয় থমিটেক (stack এ য সবেচেয় উপের

৩০

িছল) ি তীয়িটর ভতের ঢুিকেয় িদেত হেব। যমন মেন কর *A* = *{{},{{}}}* stack এর উপের আেছ আর *B* = *{{},{{{}}}}* stack এ উপর থেক ি তীয় ত আেছ। তাহেল এেদর union হেব *{{},{{}},{{{}}}}*, intersection হেব *{{}}* আর add হেব *{{},{{{}}},{{},{{}}}}*.

িতিট অপােরশেনর পর তামােক stack এর উপের থাকা সেটর সাইজ ি ট করেত হেব।

**সমাধান:** ও বাবা! সেটর সট! তেব এই দেখ ভয় পেয় গেল হেব না। আমােদর সমাধান করেত হেব। আমরা যা করেত পাির তাহেলা িতিট সটেক একিট কের id িদেত পাির। ফেল একিট সেটর ভতের যিদ কেয়ক ধাপ nested সটও থােক আমরা আসেল তার id িনেয় কাজ করব, সই nested সট িনেয় না। অথ াৎ িতিট সট এখন আসেল একিট id, আবার কােনা একিট সট বা id হেলা অ িকছু id এর সট বা set<int>. stack এর উপেরর সেটর সাইজই হেলা আমােদর িত অপােরশেনর পর উ র। যখনই নতুন কােনা সট তির হেব আমরা map এ দেখ নব য এই সট আেগই এেসিছল িকনা, আর না আসেল তােক একিট নতুন id িদব। অথ াৎ আমােদর map হেব map<set<int>, int> অথ াৎ এিট একিট int এর সট ক int এ map কের। এছাড়াও আমােদর আেরকিট reverse map এর দরকার হেব অথ াৎ কান id ত কান সট। তেব সজ তামরা চাইেল map না ব বহার কের একিট vector ব বহার করেত পােরা (vector<set<int>>). আর সব শেষ আমােদর একিট stack লাগেব। চাইেল তামরা সেটর stack িনেত পােরা, আবার চাইেল id এর stack িনেত পােরা। id এর সট নওয়া মেন হয় িবধা জনক।

মেন কর আমরা একিট id এর stack িনলাম। Push বলেল আমরা একিট ফাকা সট িনেয় দখব map এ ইেতামেধ ই এরকম কােনা সট আেছ িকনা, না থাকেল এেক একিট নতুন id িদেয় map এ রাখব এবং সই id আমােদর stack এ push করব। Dup বলেল আমরা stack এর উপেরর element িনেয় তােক আবার push করব। Union বা Intersect বলেল আমরা stack হেত িট id িনেয় আমােদর vector হেত দখব এরা কান কান set. এর পর সই িট সেটর মােঝ union বা intersect কের পাওয়া নতুন সটেক map, stack ও vector এ রাখব (এমনও হেত পাের য map এ ইেতামেধ ই আেছ, সে ে map বা vector এ রাখার দরকার নই)। সবেশেষ Add এর ে আমরা একিট id ক ওপর সেট ঢুিকেয় আেগর মত map, stack ও vector ক আপেডট করব। এখােন একটা িজিনস খয়াল রখ, vector এ থাকা আেগর সটেক যন পিরবত ন কের বস না। id ক সেট insert করার আেগ তামােক সই সট copy কের িনেত হেব। এসব ছাট খাট ভুল হেতই পাের। তামােদরেক debug কের কের বর করেত হেব এসব ভুল।

আমরা মাটােমািট map ও vector এর ব বহার ইেতামেধ ই দেখ ফেলিছ। এবার দখা যাক stack এর ব বহার। আিম ধের িনি তামরা stack নামক ডটা াকচার ইেতামেধ ই জােনা। এিট ব বহােরর জ আমােদর দরকার stack নামক হডার ফাইল। একিট int এর stack ক declare করেত হয় stack<int> S এরকম কের। এর দরকাির ফাংশন িল হেলা push(x) - তাহেল x পুশ হেয় যােব। pop() - তাহেল উপেরর element মুেছ যােব। খয়াল কর pop িক িকছু return কের না, কবল উপেরর element ক ফেল দয়। top() - stack এর উপেরর element ক return করেব তেব এিট মুেছ যােব না। empty() - এিট true দেব যিদ তামার stack ফাঁকা হয়। size() - এিট stack এর সাইজ বলেব।

UVa 540 Team Queue

**সমস া:** অেনক িল মা ষ আেছ এবং বলা আেছ ক কান দেল। এখন এেক এেক মা ষ আসেছ। ক আসেছ তা বলা আেছ। এক জন এেস যিদ দেখ তার দেলর কউ আেগ থেকই লাইেন আেছ তাহেল স তােদর পছেন িগেয় দাঁড়ােব। যিদ না থােক তাহেল স সবার শেষ দািড়েয় যােব। যখন লাইন থেক কউ বর হেব তখন লাইেনর থেক বর হেব। বলেত হেব লাইন থেক ক ক িক অড াের বর হেব। মাট 200,000 এর মত instruction থাকেত পাের।

**সমাধান:** Stl এর queue ব বহার করেল বশ সহেজই সমাধান হেয় যােব। ই ধরেনর queue রাখেত হেব। একিট queue ত থাকেব লাইেন কান কান দেলর মা ষ িক অড াের আেছ (ধরা যাক এিট িবেশষ queue), আর িতিট দেলর জ আলাদা আলাদা queue. যখন একজন মা ষ আসেব তখন তার দেলর queue দেখা। যিদ সিট ফাকা নাহয় তাহেল সই queue ত তােক ঢুিকেয়

৩১

দাও। যিদ ফাকা হয় তাহেল তােক সই queue ত ঢুিকেয় দাও এবং সই সােথ তার দলেক (মােন তার দেলর id ক) িবেশষ queue ত ঢুিকেয় দাও। বর করার সময় দেখা িবেশষ queue এর েত কান দল আেছ, সই দেলর queue এর সামেনর জনেক বর কের দাও। যিদ এর ফেল সই queue ফাঁকা হেয় যায় তাহেল তার দলেক িবেশষ queue হেত বর কের ফল। তাহেলই এই সম া সমাধান হেয় যােব।

Stl এর queue ব বহােরর জ আমােদর queue হডার ফাইল লাগেব। int এর queue ক declare করেত হয় queue<int> Q এর মত কের। এর দরকাির ফাংশন িল হেলা- push(x) - x পুশ হেয় যােব, pop() - পপ হেব িক এিট কােনা িকছু িরটান করেব না, front() - queue এর সামেনর িজিনস িরটান করেব তেব পপ হেব না, size(), empty() ইত ািদ অ া stl এর মতই কাজ করেব।

UVa 136 Ugly Numbers

**সমস া:** যসকল সংখ ার াইম ফ া র িল কবলমা 2, 3 বা 5 তােদর ugly সংখ া বেল। যমন 1, 2, 3, 4, 5, 6, 8, 9 ইত ািদ। তামােক 1500 তম ugly সংখ া বর করেত হেব।

**সমাধান:** এই সম া সমাধােনর নানা উপায় আেছ। এর মােঝ একিট উপায় তামােদর বলা যাক। মেন কর একিট সেট থম িকছু ugly সংখ া আেছ। আমরা এেদর মােঝ সবেচেয় ছাটিট িনব (এবং ঐ জায়গা থেক ঐ সংখ া মুেছ দব), এিট হেব আমােদর থম ugly সংখ া। এর পর এেক 2, 3 ও 5 িদেয় ন কের ঐ সেট এই নতুন সংখ া িল রাখব। এরপর আবার সবেচেয় ছাট সংখ ািট িনব, এিট হেব আমােদর ি তীয় ugly সংখ া। আবার আমরা তােক 2, 3 ও 5 িদেয় ন কের এই নফল িলেক সেট রেখ দব। এভােব চলেত থাকেব 1500 বার। একটু িচ া করেল বুঝেব থেম সেট ধু 1 থাকেলই চলেব। তরাং আমােদরেক সট হেত িতবার minimum সংখ া বর করেত হেব। stl এর set এ এই কাজ স ব বা তামরা চাইেল priority queue ব বহার করেত পােরা। Priority queue ব বহােরর সম া হল এখােন একই সংখ া একািধক বার ঢুকােল একািধক বার ঐ সংখ া থাকেব। তাহেল িক আমােদর এই সম ায় ইবার 6 চেল আসেব (একবার 2 *×* 3 আেরকবার 3 *×* 2).১ তরাং ভাল হয় যিদ আমরা এখােন set ব বহার কির। set স েক মেন হয় ইেতামেধ ই অেনক িকছু বেলিছ। িক একটা িজিনস বিল নাই কীভােব এ থেক সবেচেয় ছাট সংখ া পােব। সহজ "\*S.begin()" ( খয়াল কর েত একিট star আেছ)। সেটর সবেচেয় ছাট সংখ া পেয় গেল এর পেরর কাজ হল সট থেক সই সংখ া মুেছ ফলা। x ক মুেছ ফলেত চাইেল আমরা িলখেত পাির "S.erase(x)" এবং এজ লগািরদিমক সমেয়র েয়াজন। বািকটু আমার মেন হয় তামরা কের ফলেত পারেব।

UVa 1592 Database

**সমস া:** একিট টিবল দওয়া আেছ। টিবেলর সেব া 10,000 িট row এবং 10 িট column থাকেত পাের। টিবেলর িতিট সেল িকছু text আেছ। িতিট row ত একিট column এর text তার পােশর column এর text হেত একিট কমা ারা পৃথক থাকেব। বলেত হেব এমন িট row (*r*1, *r*2) ও িট column (*c*1, *c*2) আেছ িকনা যন (*r*1*,c*1)=(*r*1*,c*2) ও (*r*2*,c*1)=(*r*2*,c*2) হয়। অথ াৎ ঐ সল িলর শ একই হয় িকনা।

**সমাধান:** িলিমট অ সাের তুিম চাইেলই *r*1*,r*2*,c*1*,c*2 এর চারিট লুপ চালােত পারেব না। যেহতু কলােমর িলিমট বশ কম তাহেল কলােমর িট লুপ চালােয় ফেলা। এেত কের আমােদর *c*1*,c*2 িনিদ হেয় যােব। এখন কথা হেলা এমন িট *r*1*,r*2 পাওয়া যােব িকনা যা আমােদর condition পূণ কের। আমরা যা করেত পাির তাহেলা িতিট row ত িগেয় আমােদর িনব ািচত িট কলােমর string িনেয় তােদরেক জাড়া আকাের set বা map এ রাখেত পাির। যিদ সব row দখার পর দখ একই

১যােদর এই কথা বুঝেত সম া হে - অথ াৎ কন ইবার 6 আসেব তারা হােত হােত simulate কের দখ।

৩২

জাড়া একািধক বার এেসেছ তার মােন এরকম *r*1*,r*2 পাওয়া যােব। যেহতু আমােদর ধু াঁ না বলেলই চলেব না, *r*1*,r*2 ও িদেত হেব, সেহতু আসেল আমােদর set িদেয় কাজ চলেব না। একিট map িনেত হেব আর িলেখ রাখেত হেব য এই string জাড়া অমুক row ত এেসিছল। যখন আমরা নতুন row ত এেস নতুন string জাড়া পােবা তখন আমরা map এ দখেবা এর আেগ এই জাড়া এেসিছল িকনা বা আসেলও কাথায়। এভােবই আমরা আমােদর দরকাির *r*1*,r*2 পেয় যাব।

এখন কােড আসা যাক। থম আমরা string জাড়া বানাব কীভােব? তামরা চাইেল একিট vector িনেয় তােত string িল রাখেত পােরা। অথবা চাইেল একিট structure ক declare কের তােত আমােদর িট string রাখেত পােরা। এর পর সই structure ক map করেত পােরা। তেব সে ে সম া হেলা তামােদর সই structure এর comparison function িলখেত হেব। এজ তামােদর < operator িট overload করেত হেব। এ িবষেয় তামরা নেট অথবা " া ািমং িতেযািগতা" বইেয় দেখ িনেত পােরা। চাইেল অপােরটর ওভারেলাড না কের functor ও declare করেত পােরা। তুিম যিদ vector ব বহার করেত তাহেল কােনা operator overload এর দরকার হত না, কারণ string এবং vector জেনরই < operator ভাল মতই define করা আেছ।

তেব এই সম ার ে আমার কােছ িবধা জনক উপায় মেন হয় string এর pair নওয়া। Pair হেলা stl এর একিট িজিনস। এেক আসেল ই member variable ওয়ালা একিট structure ক না করেত পােরা। এিটর জ দরকার utility নামক হডার ফাইল। এিট declare এর উপায় হেলা pair<string, string>. যিদ একিট variable এ তুিম একিট pair রাখেত চাও তাহেল তুিম এরকম িলখেত পােরা "pair<string, string> P = make pair(A, B);" যখােন A ও B িট string. কােনা একিট pair এর থম element ক ব বহার করার জ িলখেত হয় "P.first" আর ি তীয়িটেক ব বহােরর জ "P.second". Pair এর িবধা হেলা এই ে তুিম যিদ pair এর vector বা array িনেয় তােদর sort করেত চাও তাহেল কােনা অিতির comparison ফাংশন িলখেত হেব না। তেব এর সম াও আেছ। ায়ই দখা যায় য িকছু র কাড করার পর মেন হয় আহা আেরকটা variable থাকেল মেন হয় ভাল হয়। সে ে pair ক extend করা স ব হয় না। তখন দখা যায় আমােদর নতুন কের structure বানােত হয়। এর ফেল অেনক কাড পিরবত ন হয়। এখন অব tuple আেছ। তামরা চাইেল tuple ব বহার করেত পােরা। আিম খুব একটা ব বহার কির নাই এিট।

তাহেল string জাড়া represent করার ঝােমলা তা গল। িক আমােদর ইনপুট তা র কের কলাম েলা আলাদা করা থাকেব না। সেলর মান িল কমা ারা পৃথক থাকেব। তােদর আলাদা করা যায় কীভােব? বুি হেলা string এর push back ফাংশন ব বহার করা। gets িদেয় ইনপুট িনেয় (কারণ লাইেন space থাকেত পাের) এর পর কমা না পাওয়া পয push back ফাংশন ব বহার কের string বানােত থােকা। কমা পেল বা লাইন শষ হেয় গেল এই string ক ধের একিট vector এ ঢুিকেয় দাও। তাহেল েসস শেষ একিট row এর িতিট column এর মান আলাদা আলাদা কের vector এ পেয় যােব।

UVa 12100 Printer Queue

**সমস া:** একিট queue ত অেনক িল কাজ আেছ। িতিট কােজর একিট priority আেছ। তুিম queue হেত থম কাজ িনেব যিদ দেখা এর থেকও বিশ priority ওয়ালা কাজ queue ত আেছ তাহেল এেক সবার পছেন রাখেব। আর যিদ এরকম কাজ না থােক তাহেল তুিম এই কাজ করেব। Queue ত একিট কাজ mark করা আেছ। বলেত হেব সই কাজিট কখন হেব। যেকােনা কাজ করেত 1 িমিনট সময় লােগ, আর queue হেত কােনা কাজ িনেত বা রাখেত কােনা সময় লােগ না।

মাট কাজ মা 100 িট হেত পাের।

**সমাধান:** আমােদর জানেত হেব য বত মােন সবেচেয় বিশ কত priority এর কাজ আেছ। এ জ

তামরা একিট priority queue ব বহার করেত পােরা। একিট কাজ িনেয় দখেব এর priority আর priority queue এর সবেচেয় বড় সংখ া সমান িকনা। সমান হেল এই কাজিট করেব, এবং করা শেষ queue ও priority queue জায়গা হেতই এিট ফেল দেব। আর না হেয় থাকেল queue এর শেষ এই কাজ রেখ দেব। যেহতু মা 100 িট কাজ সেহতু আমােদর এই O(*n*2+*n*log*n*)

৩৩

এর অ ালগিরদেম কােনা সম া হেব না। O(*n*2) কারণ- হয়েতা িতবার সব কাজ দখার পর সব শেষরটা আমরা নব। তাহেল O(*n*) বার O(*n*) সাইেজর অ াের বা queue আমােদর ঘঁেট দখেত হে । আর িতবার priority queue থেক একিট সংখ া ফলেত সময় লােগ O(log*n*). তরাং n িট িজিনস ফলেত সময় লাগেব O(*n*log*n*). priority queue এর উপেরর সংখ া কত এই

ে র উ র পেত আসেল constant সময় লােগ।

একিট িজিনস, priority queue ব বহােরর জ তামােদর লাগেব queue নােমর হডার ফা- ইল। এছাড়া সব কাজ queue এর মত, ধু মা এর front() ফাংশন না থেক থাকেব top() নামক ফাংশন। এিট সবসময় এর ভতের থাকা সেব া মানিট top এ রােখ। এই সম ার ে িক set ব বহার করেত যেয়া না। কারণ set এ একিট সংখ া একািধক বার insert করেলও সিট মা এক- বার থােক। তামরা চাইেল set এর পাশাপািশ একিট map িনেয় তােত সংখ া িলর count রাখেত পােরা িক তােত মেন হয় ঝােমলা বিশ হেয় যায়। এর থেক priority queue ই ভাল।

৩.১ অ শীলনী

৩.১.১ সম া

SimpleX UVa 1593 Alignment of Code X UVa 1594 Ducci Sequence

X UVa 10935 Throwing cards away I X UVa 10763 Foreign Exchange X UVa 1595 Symmetry X UVa 230 Borrowers X UVa 1596 Bug Hunt X UVa 822 Queue and A X UVa 12504 Updating a Dictionary X UVa 511 Do You Know the Way to San Jose? X UVa 400 Unix Is

EasyX UVa 1598 Exchange X UVa 12333 Revenge of Fibonacci

X UVa 212 Use of Hospital Facilities X UVa 207 PGA Tour Prize Money X UVa 814 The Letter Carrier's Rounds X UVa 221 Urban Elevations X UVa 1597 Searching the Web

৩.১.২ িহ ট

**UVa 1594:** কােনা একিট sequence এর আেগ এেসিছল িকনা এটা বুঝার জ তামরা চাইেল vector এর map রাখেত পােরা। অথবা তামরা ধর 200 বা 500 বার চ া করবা য শূ পেয় যাও িকনা, পেল তা হেয়ই গল। আর না হেল বেল দাও য হেব না। চাইেল বেস বেস মাণ করেত পােরা কন 200 বা 500 যেথ !

**UVa 10763:** এই সম া সমাধােন তামরা pair এর িট vector িনেত পােরা। একিটেত সাজা pair রাখেব আেরকিটেত উলেটা pair (a,b এবং b,a). এখন িট vector ক sort কের তােদর == িদেয় compare কের দেখা। তাহেলই হেয় যােব।

**UVa 207:** তামরা তা %s িদেয় ইনপুট আউটপুট কেরছ, িক %20s বা %9.2lf বা %-20s এরকম ব বহার কেরছ? না জানেল একটু নেট দেখ জেন নাও। এই সম া stl এর অধ ােয় রাখার কারণ হেলা sort এর জ comparison ফাংশন িলখেত শখা।

**UVa 814:** এই সম ায় একিট িজিনস িশখেল মেন হয় ভাল হয়। তাহল string এর find ফাংশন। এেক তুিম যিদ কােনা একিট character দাও তাহেল তামােক বেল দেব থেম সই character

৩৪

কাথায় আেছ। এছাড়াও এর আেরকিট দরকাির ফাংশন আেছ, তাহেলা substr(x, y)- এিট x হেত y - 1 পয substring দয়। আশা কির এই িট ফাংশন ব বহার কের তামরা email address ক ই অংেশ ভে ফলেত পারেব। না পারেলও সম া নাই, শষ আ য় for loop তা আেছই।

৩৫

৩৬

অধ ায় ৪

Mathematics স িক ত সম া

UVa 11582 Colossal Fibonacci Numbers!

**সমস া:** Fibonacci সংখ ার সং া তা জানাই আেছ তামােদর? *f*(0) = 0*,f*(1) = 1*,f*(*n*) = *f*(*n −* 1) + *f*(*n −* 2). তামােদর a, b, n (0 *≤ a, b <* 264*,*1 *≤ n ≤* 1000) দওয়া থাকেব। বলেত হেব *f*(*ab*) mod *n* এর মান কত।

**সমাধান:** তামরা িট উপােয় আগােত পােরা। একিট হেলা িবিভ nএর জ থম৫০ িট*f*(*i*) mod *n* ি টকের দখা আর আশা করাযিদ কােনািহ ট পেয় যাও।আর আেরকটা উপায় হেলা theoretically িচ া করা। যারা অিভ কে টে ট তারা এই সম া দেখ থেম যা ভাবেব তাহল *ab* অেনক অেনক বড় মান। তরাং হয় আমােদর matrix exponent এর মত িকছু করেত হেব নাহেল আমােদর sequence ক সাইেকেল ফলেত হেব। যিদ কউ খুব recent সমেয় matrix exponent টক- িনক িশেখ থােকা তাহেল তা কােনা কথাই নই, স সানে এটােক সই ভােব সমাধান করেত লেগ যােব। িক একটা িজিনস, তামােক বর করেত হেব *ab* তম fibonacci সংখ া আর *ab* িক অেনক অেনক বড় সংখ া। Matrix exponent প িতেত *i* তম fibonacci সংখ া বর করেত সময় লােগ মাটােমািট log (*i*), তাহেল *ab* তম fibonacci সংখ া বর করেত লাগেব log(*ab*) = *b*log(*a*) সময়। যা অেনক অেনক বিশ হেয় যায়। তরাং matrix exponent িদেয় হেব না। তাহেল দখা যাক এই sequence ক সাইেকেল ফলা যায় িকনা। n = 6 এর জ *f*(*i*) mod *n* এর মান িল দখা যাক- 0, 1, 1, 2, 3, 5, 2, 1, 3, 4, 1, 5, 0, 5, 5, 4, 3, 1, 4, 5, 3, 2, 5, 1, 0, 1. আমরা িক আমােদর সাইেকল পেয় গিছ, কারণ আমরা পর পর 0 আর 1 পেয়িছ, অথ াৎ আমােদর সাইেকলটা আবার থেক হেয় গেছ। এ অব ায় আমােদর মেন কেয়কটা ে র উদয় হেত পাের। থমত, সবসময় সাইেকেল পড়েব িক না। ি তীয়ত, সাইেকেল পরেল িক এটা থেক সা- ইেকেল পড়েব নািক মাঝা মািঝ কাথাও থেক। তৃতীয়ত, সাইেকেলর length িকরকম হেব। আমারা এেক এেক এসব ে র উ র পাবার চ া কির।

থম - সবসময় সাইেকেল পড়েব িকনা। াঁ পড়েব। কন? কারণ আমরা সবসময় mod*n* িনি , তরাং আমােদর সংখ া িল সবসময় 0 হেত *n −* 1 পয হেত পাের। তুিম যিদ পাশাপািশ িট সংখ া নাও তারা মাট *n*2 রকম হেত পাের। অথ াৎ *n*2 + 1 length পয যাবার পর যই pair পাবা সটা অব ই এর আেগ এেসেছ কারণ *n*2 এর বিশ রকেমর pair হওয়া স ব না। তরাং আমরা সবসময় সাইেকল পাব। ি তীয় - এই সাইেকলিট িক েত পড়েব িকনা অথ াৎ 0*,*1*...,*0*,*1*,...* এরকম হেব িকনা। াঁ। কন? ধের নাও য এিট 0, 1 এ পের না, এিট মােঝর কাথাও *a, b* ত পের অথ াৎ *f*(*i*) mod *n* হেব 0*,*1*. . . , a, b, . . . , a, b, . . .*. এখন ি তীয় a, b এর কথা িচ া কর। এর আেগর সংখ া কত? b - a. িক থম a, b এর আেগর সংখ াও b - a. অথ াৎ আমােদর সাইেকলটা আসেল a, b ত িগেয় পের না, তার আেগ িগেয় পের। িক এভােব তা তুিম একঘর একঘর কের িপছােত থাকেতই পােরা তাই না? এর মােন আসেল সাইেকলটা 0, 1 থেকই হয়। এখন তৃতীয় - সাইেকলর length কত। থম ে র উ র থেক দেখছ য *n*2 এর

৩৭

মােঝ সাইেকেল পড়েবই। িক আসেল practically এর অেনক আেগই সাইেকেল পের। এমনিক n এর মান 1000 এর মােঝ 3000 ধােপর মােঝই সাইেকেল পের যায়। িক ভােব বর করলাম? সহজ, একটা কাড কেরিছ 7-8 লাইেনর।

তাহেল আর িক। িতিট n এর জ তামরা পুেরা সাইেকল বর কের আলাদা আলাদা vector এ রেখ দাও (vector এর অ াের)। এবার ইনপুট পেল দেখা এই n এর জ সাইেকেলর length কত।ধরা যাক c. তাহেল*ab* mod *c* বর করেত হেব।এটােতা bigmod ছাড়া িকছুই না। ধু খয়াল রাখবা overflow যন না হয়। তাহেল আমরা *f*(*ab*) mod *n* এর মান পেয় যাব যা vector এর *ab* mod *c* তম সংখ া।

UVa 12169 Disgruntled Judge

**সমস া:** একটা ু জাজ মেন মেন চারিট সংখ া *x*1*, a, b, T* িনেলা (0 *≤ x*1*, a, b ≤* 10000). এর- পর স *xi* = (*axi−*1+*b*) mod 10001 ব বহার কের *x*2*,x*3*,...x*2*T* এর মান বর করল। এরপর স তামােক *x*1*,x*3*,x*5*,...x*2*T−*1 এর মান িল িদল আর তামােক বলল য *x*2*,x*4*,x*6*,...x*2*T* এর মান ি ট করেত। যিদ একািধক উ র থােক তাহেল যেকােনা একিট িদেলই চলেব। একািধক উ র মােন এমনও তা হেত পাের য একািধক *a, b* এর জ একই *x*1*,x*3*,x*5*,...x*2*T−*1 িক িভ *x*2*,x*4*,x*6*,...x*2*T* পাওয়া যায়। সে ে যেকােনা একিট িদেলই চলেব। T এর মান সেব া 100 হেত পাের।

**সমাধান:** আমরা যিদ a ও b এর উপর লুপ চালাই এবং 2*T* িট সংখ া বর কের দিখ য তারা সমান িকনা তাহেল একটু বিশ সময় লেগ যায় *a×b×T* = 1010. তাহেল কীভােব আগােনা যায়? তুিম ই ভােব িচ া করেত পােরা। এক- a ও b ইিট লুপ না চািলেয় একিট লুপ চালােনা যায় িকনা। ই- আমােদর *x*1 ও *x*3 দওয়া আেছ। এখান থেক কােনা ভােব a ও b এর মােনর ব াপাের কােনা সাহায পাওয়া যায় িকনা।

থেম *x*3 ক *x*1 এর সােপে িলেখ দখা যাক- *x*3 *≡ ax*2 + *b ≡ a*(*ax*1 + *b*) + *b ≡ a*2*x*1+*ab*+*b*( mod10001) বা *x*3*−a*2*x*1 = *b*(*a*+1)+10001*c* যখােন c একিট integer১. যিদ আমরা a এর উপর লুপ চালাই তাহেল এই equation এ ইিট unknown থাকেব b আর c. তাহেল আমরা আমােদর equation ক িলখেত পাির- *bp*+*cq* = *r* যখােন *p,q,r* constant এবং b এর মান সেব া 10000 হেত পাের। কীভােব আমরা এরকম ভােব িলখলাম? সহজ, আমরা আেগর সমীকরেণ *p* = *a* + 1, *q* = 10001 আর *r* = *x*3 *− a*2*x*1 বসাই তাহেলই আমরা *bp* + *cq* = *r* পেয় যাব। এখন আমরা b এর এমন সমাধান চাই যন তা অঋণা ক ও সেব া 10000 হয়। আমরা সাধারনত unknown ক x, y, z ইত ািদ ারা কাশ কের থািক। তাই আমরা িবধার জ b ক x ও c ক y ারা পিরবত ন কির। তাহেল আমােদর সমীকরণ দাঁড়ায় *px* + *qy* = *r*.

আমরা িক এই ধরেনর সমীকরণ সমাধান করা জািন। আমােদর euclid এর অ ালগিরদম ব ব- হার করেত হেব বা extended gcd খাটােত হেব। সে ে আমরা এমন *x*0*,y*0 পােবা যন *px*0 + *qy*0 = *g* যখােন g হল p ও q এর gcd. এর general solution হেব *p*(*x*0+*tq*)+*q*(*y*0*−tp*) = *g* যখােন t হলinteger constant. এটা আসেল extended gcd এরই একিট properties. এখন আমােদর ডান পাশেক r এর সমান করার জ ই িদেক আরও একিট integer constant ধরা যাক k িদেয় ন করেত হেব (*k* = *r*/*g*). তাহেল দাঁড়ােব *kp*(*x*0 + *tq*) + *kq*(*y*0 *− tp*) = *r*. আমরা বেলিছলাম য আমরা এমন অঋণা ক b (যা পরবত ীেত x হেয় িগেয়িছল) চাই যা এই equation ক satisfy করেব এবং 10001 এর থেক ছাট হেব (বার বার 10000 এর সমান বা ছাট বলার থেক 10001 এর থেক ছাট বলা সহজ)। অথ াৎ 0 *≤ k*(*x*0 + *tq*) *<* 10001 বা *−x*0/*q ≤ t <* 10001/*qk−x*0/*q*. একটু উপের যিদ তাকাও তাহেল বুঝেত পারেব *q* = 10001. তার মােন *−x*0/10001 *≤ t <* 1/*k − x*0/10001. এখােন খয়াল কর 1/*k* িক খুবই ছাট সংখ া, [*−x*0/10001*,*1/*k − x*0/10001) এই বাউে ডর ভতের আসেল একিটর বিশ integer থাকেব না। এর মােন এত ণ এেতা খেটখুেট এেতা এেতা constant, এেতা এেতা equation

১*a ≡ b*(mod*c*) হেল আমরা িলখেত পাির *a* = *b* + *kc* যখােন k একিট integer.

৩৮

সমাধান কের আমরা জানেত পারলাম য, িতিট a এর জ আমরা আসেল সেব া 1িট b পােবা

যটা আমােদর *x*1 হেত *x*3 িদেত পারেব।

তাহেল সারমম দাঁড়াল- আমরা a এর উপর লুপ চালােবা। Extended gcd ব বহার কের b এর মান বর করব। এরপর একিট T এর লুপ চািলেয় *x*1 *...x*2*T* বর কের িমিলেয় দখেবা য এরা আমােদর ইনপুট এর সােথ মেলিকনা।আমােদর সময় লাগেব *a×T* = 106. সােথextendedgcd এর একটা ফ া র আেছ, আমরা সটােক গণায় ধরলাম না। আর আসেল আমােদর সকল *x*1 *...x*2*T* এর মান বর না করেলও চেল। যখন কােনা একিট মান ম াচ করেব না তখন আমরা পেরর a এর মান পরী া করব। এর ফেল সময় আরও কম লাগেব।

UVa 10375 Choose and divide

**সমস া:** তামােক p, q, r, s দওয়া আেছ যখােন এেদর সবাই অঋণা ক ও সই সােথ সেব া 10000. সই সােথ *p ≥ q* ও *r ≥ s*. তামােক (*pq*)/(*rs*) বর করেত হেব। এটা বলা আেছ য উ র 108 এর বিশ হেব না এবং তামােক 5 digit precision পয ি ট করেত হেব।

**সমাধান:** আিম জািন না জােজর ট কস কতটু শি শালী। আিম এই বেলম পেল িক িক সাবধানতা অবল ন কের সমাধান করতাম তা দখাই। থমত এটা খয়াল করেত হেব য উ র ছাট হেব মােন এই না য (*pq*) বা (*rs*) ছাট। এমন হেত পাের য ই জনই অেনক বড় িক ভাগ করেল ছাট হেয় যায়। আবার সই সােথ যেহতু উ র আমােদর decimal এ িদেত হেব তরাং আমােদর খুব perfectly এেদর মান বর করেত হেব সরকম কথাও নই। তরাং আমােদর মাটােমািট সাবধানতা অবল ন করেলই চলেব। সাবধানতা কন? মেন কর তুিম থেম (*rs*) বর কেরছ এটা অেনক বড়, তরাং তামার precision loss হেয় িগেয়েছ double এ রাখেত।১ ফেল তুিম যখন আেরকিট বড় সংখ া (*pq*) ক ভাগ করবা তখন সই আেগর precision loss তামােক ভাগােব। মােন 101 ক 100 িদেয় ভাগ করা আর 100 ক 100 িদেয় ভাগ করা তা আলাদা িজিনস তাই না (যিদ কি উটার মা ২ ঘর precision রােখ)? তাহেল িক করা যায়? থমত আমরা আমােদর term ক একটু ভে িলিখ। (*p−q*+1)*...p*

1*...q ×* 1*...s*

(*r−s*+1)*...r*. অথ াৎ উপের এক সংখ া আেছ, িনেচ এক সংখ া আেছ। উ র 108 এর বিশ হেব না, আর এেদর কােনা সংখ াই 104 এর বিশ হেব না।

তুিম যা করবা তাহেলা একিট variable িনবা ধরা যাক ans. আর L আর R িদেয় ans এর জ lower এবং upper bound িঠক করা হয়। এই L আর R িক তা একটু পেরই পির ার হেব। যাই হাক, ans এর মান েত 1 কারণ আমরা এখন এেক এেক সব সংখ া ন ভাগ করব। এখন দেখা য আমােদর উ র L এর থেক ছাট িকনা, ছাট হেল এবং উপের যিদ এখেনা সংখ া বািক থােক তাহেল এেক উপেরর একিট সংখ া িদেয় ন করবা। আর যিদ ans এর মান R এর থেক বড় হয় এবং িনেচ যিদ এখেনা সংখ া থেক থােক তাহেল িনেচর কােনা একিট সংখ া ারা ভাগ করব। আর আেগর িট condition যিদ পূণ না হয় তাহেল আর িক, বািক যেকােনা সংখ া িদেয় ন ভাগ করেলই হেব। যিদ তামরা L = 1 আর *R* = 1012 নাও তাহেলই হয়েতা হেয় যােব। এখােন কােনা ধরা বাধা িনয়ম নাই। মুল আইিডয়া হল আিম আমার ans ক কখনও খুব বড় হেত দব না আবার খুব ছাট হেত দব না। তুিম চাইেল এই েসেসর েতই উপেরর সব সংখ ােক একিট vector আর িনেচর সব সংখ ােক আলাদা আেরকিট vector এ রাখেত পােরা। অথবা যেহতু উপর িনচ জায়গােতই সংখ া িল িট েপ আেছ সেহতু িট variable রেখও তুিম র বা সংখ া সামলােত পােরা।

১তুিম এভােব িচ া করেত পােরা য, কি উটার double এ কেয়ক ঘর রাখেত পাের মা । এখন যিদ তুিম এখােন অেনক বড় মান রাখেত যাও তাহেল কি উটার শেষর িদেকর অংক আর রাখেব না, সখােন স 0 বিসেয় দেব।

৩৯

UVa 10791 Minimum Sum LCM

**সমস া:** তামােক N (*<* 231) দওয়া থাকেব। তামােক এমন িকছু সংখ া (একািধক হেত হেব) বর করেত হেব যন তােদর লসা N হয় এবং সই সংখ া িলর যাগফল সব িন হয়। যমন N = 12 হেল আমােদর সংখ া িল হেব 3 আর 4. কারণ এেদর লসা 12 এবং এেদর যাগফল 7 যা সব িন । তামােক সই সব িন যাগফলিট ি ট করেত হেব। মাট 100 িট test case দওয়া থাকেব।

**সমাধান:** তামােক যিদ িকছু সংখ া দওয়া থােক তাহেল তােদর লসা কীভােব বর করেব? অেনক উপােয়ই বর করা যায়। তার মােঝ একিট উপায় হেলা সংখ া িলেক prime factorization করা। এরপর িতিট prime এর জ দখা য তামােক দওয়া সংখ ােদর মােঝ কােনা সংখ ায় তার সবেচেয় বড় power কত আেছ। এভােব িতিট prime এর জ সই বড় power িদেয় ন করেলই তুিম লসা পেয় যােব। উদাহরণ দওয়া যাক। ধর আমরা 4, 6, 18 এর লসা বর করেত চাই। এখন 4=22, 6=2 *×* 3 আর 18 = 2 *×* 32. তাহেল 2 এর সবেচেয় বড় power আেছ 2, আর 3 এর সবেচেয় বড় power আেছ 2। তাহেল আমােদর লসা হেব 22 *×* 32 = 36. এর মােন যিদ আমােদর লসা দওয়া থােক তাহেল এেক আেগ prime factorize কের নই। যেহতু লসা খুব জাড় 231 হেত পাের আমরা খুব সহেজই *√N* এ তার prime factorize করেত পাির। বা তােত যিদ খুব বিশ সময় লেগ যায় তাহেল আমরা *√N* পয prime generate কের রেখ N ক prime factorize করেত পাির। ধরা যাক এই prime factorization িট হল *pk*11 *pk*22 *pk*33 *...*. এর মােন দাঁড়াল আমরা যই সংখ া িল িনব াচন করব তােদর িতিটেত *p*1*,p*2 *...* এর বাইের কােনা prime থাকা যােব না, আর এেদর power আবার যথা েম *k*1*,k*2 *...* এর বিশ হওয়া যােব না। কীভােব তুিম এই prime িলেক সংখ া িলর মােঝ distribute করেল সংখ া িলর যাগফল সব িন হয়?

তামরা চাইেল এখােন একটু িবরিত িনেয় িচ া করেত পােরা।

থমত, কােনা একিট prime একািধক সংখ ায় থাকেব না। কারণ একািধক সংখ ার মােঝ কােনা একিটেত prime িটর সেব া power থাকেব। সিট বােদ বািক িল মুেছ িদেল আমােদর সংখ া িলর যাগফল আরও কেম যােব তাই না? একটা উদাহরণ দখা যাক। ধর 12 = 22 *×* 3 তাহেল কােনা একিট সংখ ায় 22 থাকেব। তাহেল আমরা চাইেল একিট 22 রেখ বািক 2 সব মুেছ ফলেত পাির। আবার 22 আর 3 চাইেল একই সংখ ােতও থাকেত পাের বা িভ সংখ ােতও থাকেত পাের। যমন 22*,*3 হেত পাের বা 223 হেত পাের। কানটা লাভজনক? অব ই 22 + 3 *<* 223. একটু িক বুঝেত পারছ কীভােব prime িলেক রাখেল যাগফল সবেচেয় কম হেব? াঁ তােদর আলাদা আলাদা কের রাখেত হেব। অথ াৎ *pk*11 *,pk*22 *,...* এরকম। কন? খয়াল কর এখােন মুল ব াপার হল *ab* ছাট নািক *a*+*b*? দখা যাক, *ab*?*a*+*b ⇒ ab−b*?*a ⇒ b*? *aa−*1. এখােন ? মােন হল আমরা জািন না য এখােন িক বসেব <, > নািক =. আমােদর ডান পাশ ায় 1 এর কাছাকািছ একটা সংখ া। আর বাম পাশ তা কমপে 2. এর মােন সবসময় *ab* সংখ া *a* + *b* এর থেক বড় হেব যখােন a, b িতিট prime এর power (আমােদর এ সম ার ে এিট সত , তাই বেল সব ে *ab > a* + *b* না যমন 1 *∗* 1 *>* 1+1 িক সিত না)। অথ াৎ আমােদর optimal সংখ া িল হেব *pk*11 *,pk*22 *,...*.

সব শেষ একটা িজিনস, আশা কির খয়াল কেরছ য তামােদর সেট একািধক সংখ া থাকেত হেব। তরাং যিদ দ N এ মা একিট াইম থােক তাহেল িক করবা সটা িনেজরা িচ া কের বর কর।UVa 12716 GCD XOR

**সমস া:** মাট 10,000 িট ট কস। িত কেস তামােক N (*≤* 3 *×* 107) দওয়া থাকেব। তা- মােক বলেত হেব এমন কত িল (*A, B*) আেছ যন 1 *≤ B ≤ A ≤ N* হয় ও *gcd*(*A, B*) = *xor*(*A, B*) হয়। যমন *N* = 7এর জ উ র 4 এবং (*A, B*) িলহল (3*,*2)*,*(5*,*4)*,*(6*,*4)*,*(7*,*5).

**সমাধান:** িট সংখ ার gcd এর সােথ xor এর কানই স ক আপাত দৃি ত দখা যাে না। আমরা যসব প িতেত gcd বর কির তার সােথ xor তা দূেরর কথা bitwise representation

৪০

এরই কান স ক দখা যাে না। এরকম অব ায় যটা করা উিচত সটা হেলা pattern খাঁজা। একটা ছাট কাড কের N এর মান 100 পয যসব pair পাওয়া যায় তােদর ি ট কের ফেলা। িক এরপরও কােনা pattern পাওয়া যায় না। যমন িকছু িবদঘুেট pair হেলা (85*,*68), (90*,*72) ইত ািদ। যারা আসেলই pattern খাঁজায় পটু তারা হয়েতা ইেতামেধ ই pattern পেয় গেছ। যিদ খয়াল না কের থােকা তাহেল বিল এেদর gcd বা xor হল তােদর মােঝর পাথ েক র সমান। যিদও এটা িঠক য যেকােনা pair এর পাথ ক সবসময় তার gcd হয় না বা xor ও হয়না। তেব যিদ তােদর xor ও gcd সমান হয় তাহেল তােদর পাথ ক ও এেদর সমান হয়। অ ত N এর মান সেব া 100 এর জ যত িল pair পেয়ছ তােদর সবার ে এটা সত । এখন িট িজিনস। থমত- এটা িক আসেলই সত ? মাণ িক? ি তীয়ত- এটা কীভােব efficiently কাড করা যােব।

অেনেক থম এর উ র না বর কেরই ি তীয় অংেশ চেল যায়। অথ াৎ মাণ না কেরই কাড কের বা কাড কীভােব করেব তা িচ া কের। া ািমং কে ট এর সময় মাণ খুব একটা দরকাির না। তুিম আরও িনি ত হেত হয়েতা N এর মান 1000 বা 10,000 পয চক কের দখেত পােরা। যিদ এেতা পয তামার claim সিত হয় তাহেল ধের নয়া যায় য আসেল সব N এর জ তামার claim সত হেব। যিদও যত ণ না মাণ করছ তত ণ মেন একটা খচখচ করেত থােক। তাহেল চল এটা মাণ করা যাক। খয়াল কর, *A − B ≤ xor*(*A, B*). কন? যিদ B ত যসব bit 1 আেছ A তও যিদ তারা 1 থােক তাহেল *A − B* আর *xor*(*A, B*) সমান হেব। অ থা xor এর মান বিশ চেল আসেব। আবার অ িদেক *gcd*(*A, B*) িক A আর B জনেকই ভাগ কের। তাই *gcd*(*A, B*) িক *A − B* ক ভাগ কের। অথ াৎ *gcd*(*A, B*) *≤ A − B*. তাহেল দখেত পা *gcd*(*A, B*) *≤ A −B ≤ xor*(*A, B*). অথ াৎ যিদ xor আর gcd সমান হয় তাহেল তারা *A − B* এর সমান হেব। িক র সহেজ মাণ হেয় গল। িক মেন রেখা এই মাণ সহজ হেয়েছ কারণ তুিম আেগ থেকই জােনা য এখােন *A − B* এর কােনা ভূিমকা আেছ। নাহেল তুিম *A − B* এর সােথ gcd বা xor এর স ক খুঁজেত যেত না। মাট কথা তামার pattern খাঁজা বা explore করার মানিসকতা থাকেত হেব।

এখন তাহেল ি তীয় ে িফের আসা যাক। কীভােব এিট সমাধান করবা। থমত িন য় তুিম A ও B িটর উপেরই লুপ চালােত পারবা না। আবার ধু A বা ধু B এর উপের লুপ চািলেয় িক লাভ হেব তাও বুঝা যাে না। এরা জন ছাড়া লুপ চালােনার মত candidate আেছ gcd. দখা যাক gcd এর উপর লুপ চািলেয় লাভ হয় িকনা। ধরা যাক আমােদর gcd এর লুেপর variable হেলা g. তাহেল আমরা A ও B স েক িক বলেত পাির? থমত A ও B উভেয়ই g এর multiple ( িণতক)। ি তীয়ত A - B হেব g এর সমান (আমরা িকছু ণ আেগ তাই মাণ কেরিছ)। এর মােন আমরা যিদ A জািন তাহেল B হেব *A−g*. আমরা যিদ A এর উপর লুপ চালােত চাই তাহেল কীভােব চালাব? একিট উপায় হেলা 1 হেত N পয i এর লুপ চািলেয় দখা য i িক g ারা িবভাজ িকনা। িক এেত িতবার O(*N*) এর লুপ চালােত হয়। তা না কের তুিম সরাসির g এর multiple এর উপর লুপ চালােত পােরা। g হেত N পয এবং িতবার লুেপর মান g পিরমাণ কের বািড়েয় [for(A = g; A <= N; A += g)] অথবা 1 হেত *N*/*g* পয i এর লুপ চালাও তাহেল A হেব *i× g*. তাহেল এই nested loop চলেত সময় লাগেব O(*N*/*g*). যেহতু g এর লুপ 1 হেত N পয চালা তাই মাট সময় লাগেব *N*/1 + *N*/2 + *...N*/*N* এবং এিট আসেল *N* log*N*. তরাং এই িট nested লুপ চািলেয় আমরা A, B আর g জেন গলাম। আমরা খুব সহেজই চক করেত পাির য *xor*(*A, B*) = *g* িকনা। িক *gcd*(*A, B*) = *g* িকনা তা চক করেত তা আরও একটা O(log*N*) লাগেব। িক একটু িচ া কের দেখা এই চক এর কােনা দরকার নই। কারণ A আর B পাশাপািশ g এর multiple. মােন *A* = *ig* হেল *B* = (*i−*1)*g* এবং এেদর মােঝ g ছাড়া আর কােনা সাধারণ ণনীয়ক নই কারণ g ছাড়া এরা হেলা *i* আর *i−*1. আর পাশাপািশ িট সংখ া সবসময় coprime. কন? মেন কর তারা coprime না। এর মােন এরা p ারা িবভাজ । এর মােন এেদর িবেয়াগফল *i−*(*i−*1) ও p ারা িবভাজ ( িট সংখ া p ারা িবভাজ হেল তােদর িবেয়াগফলও p ারা িবভাজ হেব)। মােন 1 সংখ ািট p ারা িবভাজ । তােতা স ব না তাই না?

তাহেল আমরা আমােদর সমাধান পেয় িগেয়িছ। আমরা থেম gcd এর উপর লুপ চালােবা। এর পর A এর উপর লুপ চালােবা যটা হেব gcd এর multiple. আর *B* = *A − g*. সবেশেষ চক করব *xor*(*A, B*) = *g* িকনা। শষ!

তাহেল এই সম া থেক িক িশখলাম? িশখলাম য মােঝ মােঝ ধয ধের pattern খুঁজেত হয়।

৪১

UVa 1635 Irrelevant Elements

**সমস া:** মেন কর তামার ব 5ু n আর m িট সংখ া িনব াচন কেরেছ (1 *≤ n ≤* 10*,*2 *≤ m ≤* 109). এবাের স n িট সংখ া *a*0*,a*1*,...an−*1 িনব াচন কেরেছ যখােন িতিট সংখ াই 0 হেত *m−*1 এর মােঝ। এবার স পাশাপািশ িট কের সংখ া যাগ করল। তাহেল স পল *a*0+*a*1*,a*1+ *a*2 *...an−*2 + *an−*1 এই *n −* 1 িট সংখ া। এবার আবার স একই কাজ করল এবং পল *n −* 2 িট সংখ া। এভােব স করেতই থাকল যত ণ না স একিট সংখ া পল। এবার স এই সংখ ার mod*m* িনেলা। এখন তামােক বলেত হেব শষ সংখ ার উপের কান কান *ai* এর কােনা ভাব নই। যমন *n* = 3 ও *m* = 2 হেল শষ সংখ ার উপের *a*1 এর কােনা ভাব নই। কন? কারণ *n* = 3 এর জ শষ সংখ ািট হেলা *a*0 + 2*a*1 + *a*2 mod *m* আর যেহতু তুিম 2 িদেয় mod িন সেহতু শষ সংখ ােত *a*1 এর কােনা ভাব নই। **সমাধান:** যারা অিভ তারা এই সম া পড়ার সােথ সােথই বুঝেত পারেব য শষ সংখ া িক হেব। এখােনই আসেল অিভ আর অনিভ েদর পাথ ক । অনিভ রা বশ িকছু সময় ব য় করেব এই common িজিনস বর করেত আর অিভ রা এক দখােতই বুঝেত পাের য শষ মানটা িক হেব। যিদ তামরা না জােনা য শষ মান িক হেব তাহেল n এর িবিভ মােনর জ শষ মান বর কের দখেত পােরা। যমন *n* = 2 এর জ *a*0 + *a*1. *n* = 3 এর জ এর জ *a*0 + 3*a*1 + 3*a*2 + *a*3. বুঝেত পারছ িক হে ? n এর জ *a*এই 0 + মান 2*a*1 হেব + *a*(2. *n−*1

0

*n* = 3 )(*n−*1

*a*0 + 1

)*a*1 +(*n−*1

2

)*a*2 +*...*(*n−*1

*n−*1)*an−*1. কন? কারণটা বুঝার সবেচেয় ভাল উপায় হল pascal এর triangle এর মত আঁকা এবং বর করা িতিট সংখ ার contribution কত কের। িচ ৪.১ এ িজিনসটা বুঝার চ া কের দখেত পােরা। আমরা িক ি ভুজই এঁেকিছ তেব উলেটা কের। এখােন আমােদর থম সংখ ািট হেলা উপেরর বাম কাণা এবং এর পর বািক সংখ া িল পরপর আেছ। এখন কােনা একিট সংখ ার contribution হেলা সিট যই ােন আেছ সখান থেক িনেচর কাণায় যত ভােব যাওয়া যায় তা। যমন িচে র *a*1 হেত িনেচর কাণায় বােম আর ই ধাপ ডােন যেত হেব। 1টা L আর 2টা R ক যভােবই আমরা (2+1যাও 1

) না = কন (4*−*11

তামােক ) ভােব সাজােত একধাপ

পাির (ভুেল যও না আমােদর n এখােন িক িহসাব কর তাহেল দখেব *ai* এর coefficient 4)। হেব একই (*n−*1

*i*

ভােব ).

তুিম িতিট *ai* এর জ যিদ একটু

*a*0 *a*1 *a*2 *a*3

নকশা ৪.১: Pascal Triangle

মুল সম ায় িফের যাওয়া যাক। আমােদর িতিট *i* এর জ মেন কর 1 কিমেয় িনেয়িছ) m ারা িবভাজ িকনা। এখােন m িক বর করেত হেব য prime নাও হেত (*n*) *i*পাের। (n (*n*ক ) *r*বর করার অেনক উপায় plan (*nn*) পয করেত পাির য থেম আেছ। দখেবা তামােক (*n*0) িক িবিভ m উপায় এেক এেক িচ া কের ারা ভাগ যায় িকনা, এর পর দখেবা দখেত (হেব। *n*1আিম যায় তা িচ সব া করেত িল। হেব। তরাং সহজ, আমােদরেক (*ni*) এভােব ) পেরর ক িলখেত টাম পাির েক (*n*আেগর *i*টাম এর সােপে কীভােব বর করা factorize ) = *n−i*+1

*i* কত। এবার কের এেক এেক ফেলা (এবং *ni*) বর কর এর ভতের কান কান prime ( *i−*1*n*

আেছ ). এরপর m আর তােদর ক prime power িল বর কর। এজ তামােক িতবার *n−i*+1 ন করেত হেব আর *i* ভাগ করেত হেব। একটু িচ া করেল বুঝেব ন আর ভােগর সময় তামার কবল মা m এর prime divisor েলােতই আ হ আেছ। আর অ prime েলা গা ায় গেলও কােনা সম া নই। এখন তামােক যা করেত হেব তাহেলা এই prime িলর power মেন রাখা। যিদ দেখা সব িল power

৪২

ই m এ থাকা power এর সমান বা বড় তাহেল িবভাজ নাহেল িবভাজ না।

তাহেল আমােদর সমাধান দাঁড়াল- থেম m ক prime factorize কের এর prime divisor আর তার power িল বর করব। এর পর এেক এেক (*ni*) বর করব (i এর মান 0 হেত n পয )। িতবার একিট সংখ া ন আর ভাগ করব। এই ন ভােগর সময় আসেল আমরা দখেবা m এর ভতের থাকা prime িল কতবার কের আেছ। ভােগর সময় এই count কমেব আর েনর সময় বাড়েব। ন ভাগ হেয় গেল িতিট prime এর power চক কের দখেবা য m এ থাকা power অেপ া কউ কম আেছ িকনা। যিদ কম থােক তাহেল িবভাজ না, আর যিদ সবাই বিশ বা সমান হয় তাহেল িবভাজ । n িক খুব একটা বড় না তাই তােত খুব একটা বিশ prime factor থাকেব না। তাই এই সমাধােনর complexity ও অেতা বিশ হেব না।

UVa 10820 Send a Table

**সমস া:** তামােক N (*≤* 50*,*000) দওয়া থাকেব। বলেত হেব কত িল (*x, y*) আেছ যােত কের x ও y এর মােঝ কােনা সাধারণ ণনীয়ক না থােক (1 *≤ x, y ≤ N*). মাট 600 িট test case থাকেত পাের।

**সমাধান:** x ও y এর মােঝ কােনা সাধারণ ণনীয়ক থাকা যােব না- এটা আসেল phi function বা Euler's totient function এর কথা মেন কিরেয় িদে । *phi*(*i*) বেল য i এর সমান বা ছাট কত িল সংখ া আেছ যার সােথ i এর কােনা সাধারণ ণনীয়ক থাকেব না। এর মােন আমােদর বেলেম *x ≤ y* এর জ উ র হেব ∑*Ni*=1 *phi*(*i*) কারণ আমরা িতবা *i* = *y* ক fix করিছ আর দখিছ য কত িল xপাওয়া যায়।আর *phi*(*i*) কীভােব বর করেত হয় তা া ািমংকে ট বইেয় বা অ কাথাও দেখ িনেত পােরা। যিদ আমরা phi বর করা িশেখ যাই তাহেল আমরা phi এর একটা cumulative sum এর অ াের বািনেয় রেখ িদেত পাির। Cumulative sum এর অ াের মােন হল *sum*(*i*) = *phi*(1) + *. . . phi*(*i*)। এটা তা বর করা সহজ *sum*(*i*) = *sum*(*i −* 1) + *phi*(*i*). যিদও আমরা *phi*(*i*) এ i এর থেক বড় সংখ া যারা i এর সােথ coprime তােদর পাই নাই তেব এে ে আমরা সকল *x ≤ y* এর জ সমাধান পেয় িগেয়িছ। আর বািক থােক *i>j* যা আসেল *i ≥ j* এর সমান, মােন আমরা িকছু ণ আেগ য মান বর কেরিছ তার ায় সমান। ায় কন বললাম

সটা তামরাই একটু কাগজ কলম ব বহার করেল বুঝেত পারেব।

UVa 1262 Password

**সমস া:** তামােক িট 6 *×* 5 (6 িট row, 5 িট column) সাইেজর character এর ি ড দওয়া থাকেব। 5 length এর একিট শ valid password হেব যিদ শ িটর i তম character িট িট ি েডরই i তম কলােম থােক। তামােক K দওয়া থাকেব। K তম lexicographically smallest valid password ি ট করেত হেব।

**সমাধান:** Counting সম ার ে lexicographically smallestউ র বর করার একিটcommon সম া এিট। মুল আইিডয়া হেলা তুিম বাম হেত ডােন এেক এেক উ রিটর character িল পূরণ করেব। মেন কর তুিম ইেতামেধ ই থম *i−*1 ঘর পূরণ কের ফেলছ আর এরকম অব ায় K তম শ

বর করেত চা । এখন তুিম *i* তম ঘর পূরণ করেব। তুিম থেম এখােন A বসােনার চ া কর। যিদ দেখা মাট X ভােব A বসােনা যায় তাহেল দেখা, *X ≥ K* িকনা। যিদ হয় এর মােন এই জায়গােত A ই বসেব। আর যিদ নাহয় তাহেল *K* হেত *X* িবেয়াগ কর আর i তম ােন B বিসেয় চ া কর। একই ভােব েয়াজেন C, D, E এই ভােব এেক এেক Z পয চ া কের যাও।

তাহেল কািহনী দাঁড়াল য আমরা যিদ password এর থম িকছু ঘর জানা থাকা অব ায় পেরর ঘর িলেক কতভােব পূরণ করা যায় তা যিদ বর করেত পাির তাহেল আমােদর সম া সমাধান হেয় যােব। আর এই ে র উ র মেন হয় সহজ তাই না? বািক column িলর িতিটেত কয়িট কের character বসােনা স ব তা ন করেলই পেয় যাবা।

৪৩

একটা ছাট উদাহরণ দয়া যাক। মেন কর তুিম 10ম শ বর করেত চাও। তুিম করেল িক, থম ােন A বসােল। এর পর তামার ব ুেক িজ াসা করেল, আিম থম ােন A বিসেয়িছ, তুিম িক বলেব এই অব ায় কত িল valid শ আেছ? ধর তামার ব ু বলল 1 িট। তুিম বুঝেল 10ম শ িট A িদেয় না। সই সােথ তুিম জেন গেল থম শ িট A িদেয় । তরাং এখন তামার মাথা ব াথা 9ম শ িনেয় (কারণ থম শ তা বাদ)। এখন মেন কর তুিম থম ােন B বসােল আর তামার ব ুেক িজ াসা করেল এবার বল কত িল valid শ আেছ। স বলল 50 িট। তাহেল তুিম বুেঝ িগেয়ছ য থম ােন B বসেব। এবার তামার িচ া ি তীয় ান িনেয়। তুিম এখােন A বসােল। আবার ব ুেক িজ াসা করেল য BA িদেয় শ কয়িট? তামার ব ু বলল 4 িট। তামার দরকার 9ম শ । তাহেল বুেঝ গেল ি তীয় ােন A না। এবং সই সােথ তুিম এখন থেক 5ম শে র স ান করেব। ি তীয় ােন মেন কর B বসােল আর তামার ব ু বলল BB িদেয় শে র সংখ া 0িট। তুিম এবার C িদেয় চ া কর। এভােব যিদ চািলেয় যাও তুিম দখেব পুেরা শ বািনেয় ফেলছ।

আর কয়িট শ বানােনা স ব তা বর করা বশ সহজ। থেম বর কর থম কলােম কত িল character বসেত পাের। এটা বর করা সহজ, তুিম A হেত Z িতিট character এর জ দখেব য সই character িট িট ি েডর থম কলােম আেছ িকনা। এরকম কের বর কর কত িল character িট ি েডর থম কলােম আেছ। একই ভােব তুিম িতিট কলােমর জ এই সংখ া বর কর। এেদর ন করেলই তুিম পেয় যােব য কত িল শ বানােনা স ব। যেহতু আমরা থম িদেকর িকছু কলােম character বিসেয় ফলব তরাং তােদর বােদ বািক কলাম িল িনেয় িহসাব করেলই আমরা সই বািক কলাম িল ব বহার কের কত িল শ বানােনা স ব তা পেয় যাব।

UVa 1636 Headshot

**সমস া:** একিট িপ ল আেছ। িপ লিটেত n িট িলর slot আেছ। কান slot এ িল আেছ কান slot এ নই তা বলা আেছ। এখন তামার ব ু আর তুিম russian roulette খলেব। থেম তামার ব ু িপ লিট িনেলা এবং িলর slot ঘুরাল। এর ফেল এখন িপ লিট random একিট slot এ আেছ। এই অব ায় স fire করল িক কােনা িল ব েলা না। এবার তামার পালা। তামার হােত িট option. এক- তুিম আবার slot িল ঘুরােব ফেল িপ লিট random slot এ যােব অথবা ই- তুিম কােনা রকম না ঘুরােয়ই িল করেব। তামােক বলেত হেব কান উপােয় িল বর না হবার স াবনা বিশ। যিদ ই উপােয়ই সমান হয় তাহেল বল য সমান। n এর মান 100.

**সমাধান:** আমরা ই উপােয়র স াব তা বর করব এবং তুলনা কের দখেবা য তারা সমান িকনা বা সমান না হেল কান ভােব িল না বর হবার স াবনা বিশ। থেম সহজটা বর করা যাক- তুিম যিদ slot ক ঘুরাও তাহেল িল না ব েনার স াব তা। এটা আসেল বিশই সহজ। তামার মাট n িট slot আেছ। ধরা যাক তার a িটেত িল আেছ আর *n−a* িট ত নই। তুিম যিদ slot ক randomly ঘুরাও তাহেল িল নাই এরকম slot আসার স াবনা হেলা *n−an* . কন? কারণ যেহতু n িট slot আেছ তাই randomly ঘুরােল n ধরেনর slot আসেত পাের। আর এেদর মােঝ *n − a* িটেত িল নই। তাই িল নই এরকম slot আসেত পাের *n−a* ভােব। তরাং *n−a* ক *n* িদেয় ভাগ করেল আমরা িল নই এরকম slot আসার স াবনা বর কের ফলেত পারব।

এখন চল বর করা যাক যিদ না ঘুরাও তাহেল িল বর না হবার স াবনা কত। যেহতু এর আেগর বাের তামার ব ু যখন triggure িটেপেছ তখন কােনা িল বর হয় নাই তরাং তুিম জােনা য এখন িপ ল এমন জায়গায় আেছ যার আেগর slot এ িল িছল না। তরাং তুিম খুব সহেজই নেত পােরা এরকম অব ান কয়িট আেছ। এখন হল এেদর মােঝর কয়িট অব ােনর জ পেরর অব ােনও িল নই। উদাহরণ দয়া যাক। ধরা যাক slot এর অব া 0011 (0 মােন িল নই, 1 মােন িল আেছ)। এখন তামার ব ুর ে slot থম িটর কােনা একিটেত িছল। যিদ তামার ব ুর কােছ থম অব ােন থােক তাহেল তামার ে িল হেব না। আর যিদ তামার ব ুর কােছ ি তীয় অব ােন থােক তাহেল তামার কােছ িল হেব। অথ াৎ তামার কােছ িল হবার স াবনা 0*.*5. যিদ slot এর অব া 010101 হয় তাহেল িক িল হবার স াবনা 1. কারণ তামার ব ুর ে যেহতু িল হয় নাই তরাং তামার ে িল হেবই কারণ সব 0 এর পের 1 আেছ। এটা স াব তার খুব basic সম া।

৪৪

UVa 10491 Cows and Cars

**সমস া:** একিট game show ত a িট দরজার পছেন গািড় আর b িট দরজার পছেন গ আেছ। তুিম থেম একিট দরজা িনব াচন করেল তামার জ । এর পর উপ াপক তামার দরজা ছাড়া অ দরজার মােঝ c িট দরজা খুেল দখাল য সসেবর পছেন গ আেছ (*c<b*)। এবার তুিম তামার দরজা বােদ অ যেকােনা দরজা িনব াচন করেল। বলেত হেব তামার গািড় পাবার স াবনা কত।

**সমাধান:** ধরা যাক *n* = *a* + *b* অথ াৎ মাট দরজার সংখ া n. এখন মেন কর তুিম একিট দরজা িনব াচন করেল, তাহেল এেত গািড় থাকার স াবনা *an*. বািক আেছ *n −* 1 িট দরজা। এর মােঝ *c* িট দরজা আমােদর উপ াপক খুেল দিখেয় িদেয়েছ য তােদর পছেন গ আেছ। তাহেল বািক থােক মাট *n −* 1 *− c* িট দরজা। যেহতু আিম যই দরজা িনব াচন কেরিছ তার পছেন গািড় আেছ (কারণ আমরা ধের িনেয়িছ য *a*/*n* probability ত আমােদর থম দরজার পছেন গািড় আেছ) তাই বািক *n −* 1 *− c* িট দরজার মােঝ গািড় আেছ *a −* 1 িট। তাহেল আিম যিদ বািক *n −* 1 *− c* িট দরজার একিট িনব াচন কির তােত গািড় থাকার স াবনা *a−*1

*n−*1*−c*. তরাং আমরা থম যই দরজা িনব াচন কেরিছলাম তােত এবং ি তীয় যই দরজা িনব াচন করলাম তােত গািড় থাকার স াবনা *an × a−*1

*n−*1*−c*. আবার এমনও হেত পাের য আমরা যই দরজা থেম িনব াচন কেরিছলাম তােত গ আেছ। এটা হবার স াবনা *bn*. এবার বািক *n−*1 িট দরজার মােঝ উপ াপক *c* িট দরজা খুেল গ দিখেয়িছল। বািক থােক *n −* 1 *− c* িট দরজা। এর মােঝ গািড় আেছ *a* িট। আমরা যিদ বািক *n −* 1 *− c* িট দরজা থেক randomly িনব াচন কির তাহেল গািড় পাবার স াবনা *a n−*1*−c*. অথ াৎ থম দরজায় গ আর ি তীয় দরজায় গািড় পাবার স াবনা *bn × a*

*n−*1*−c*. তাহেল গািড় পাবার মাট স াবনা দাঁড়ােব *an × a−*1 *n−*1*−c* + *bn × a n−*1*−c*.

যারা probability ত একদমনতুন তােদর মেন আসেত পাের এরকম ন কের যাগ করলাম কন। অেনক ভােবই এর ব াখ া দওয়া যায়। মেন কর তামার কােছ িতনিট coin আেছ এবং এেদর head পরার স াবনা *p*1*,p*2*,p*3. থেম তুিম থম coin টস করেব। যিদ head পের তাহেল ি তীয় coin, আর যিদ থম coin এ tail পের তাহেল তৃতীয় coin টস করেব। বলেত হেব সবেশেষ যই কেয়নিট টস করেব তােত head পরার স াবনা কত। খয়াল কর, থম coin টস করার পর িক ই রকম ঘটনা ঘটেত পাের। *p*1 স াবনায় head পরেত পাের, এবং সই ে তুিম ি তীয় coin টস করেব এবং তােত head পরার স াবনা *p*2. এই ঘটনা িট একে ঘটার স াবনা *p*1*p*2. একই ভােব আমরা বলেত পাির থম coin এ tail ও তৃতীয় coin এ হড পরার স াবনা (1*−p*1)*p*3. যখন িট ঘটনা একে ঘটােনা দরকার তখন তােদর স াবনােক ন করেত হয়। এবার আমােদর বর করেত হেব শেষর coin এ head পরার স াবনা। শেষর coin ি তীয় coin হেত পাের অথবা তৃতীয় coin হেত পাের। তরাং ি তীয় coin এ head "অথবা" তৃতীয় coin এ head পরার স াবনা হেলা *p*1*p*2+(1*−p*1)*p*3. এখােন যাগ কেরিছ, কারণ আমরা ইিট ঘটনার কােনা একিট ঘটার স াবনা বর করেত চেয়িছ। অথ াৎ "অথবা" হেল আমােদর স াব তােক যাগ করেত হয়। িক কন "এবং" হেল ন আর "অথবা" হেল যাগ? এভােব িচ া করেত পােরা- মেন কর তামার সামেন একিট dart board আেছ। তুিম সই dart board ক সমান িতনভােগ ভাগ করেল। এর পর একভাগেক সবুজ, এক ভাগ ক নীল আর আেরক ভাগেক লাল রং করেল। যেহতু িতনিট রেঙর জায়গার ফল সমান তাই তুিম যিদ একিট dart ছুেড় মার আর যিদ সটা board এর একিট random িব েত িগেয় আঘাত কের সই িব িটর রং লাল হবার স াবনা 1/3 ( ই রেঙর মােঝ পড়েল িক হেব িচ া না করেলও চলেব)। একই ভােব সবুজ হবার স াবনা 1/3, নীল হবার স াবনা 1/3. এখন যিদ বেল সবুজ "অথবা" নীল হবার স াবনা কত। িক করেব? যাগ হেব তাই না? কারণ নীল হবার স াবনা 1/3, সবুজ হবার স াবনা 1/3 তাই নীল বা সবুজ হবার স াবনা 1/3 + 1/3 = 2/3. বা অ ভােব যিদ িচ া কর তাহেল দখেব "নীল বা সবুজ" এর ফলেক যিদ তুিম মাট ফল িদেয় ভাগ কর তাহেল দখেব 2/3 পেয়ছ। তরাং অথবা হেল যাগ। এখন িচ া কর তামার কােছ এরকম িট board আেছ আর তুিম িট বােড িট dart মারেল। িটই নীল িব েত পরার স াবনা কত? 1/3 *×* 1/3 = 1/9 কন? কারণ এই িট dart 9 ভােব পরেত পাের- (লাল, লাল), (লাল, নীল), (লাল, সবুজ), এরকম আরও 6 রকম। এবং যেহতু সব িল রেঙ পরার স াবনা একই তাই এেদর যেকােনা একিট combination পাবার স াবনা 1/9. আর এেদর মােঝ একিটর ে ই িটই

৪৫

নীল। অথ াৎ 1/9. তরাং এবং হেল ন। িঠক counting এর মতই। Counting এর ে ও িক "অথবা" এর ে তুিম যাগ কর আর "এবং" হেল ন।

UVa 11181 Probability Given

**সমস া:** একিট দাকােন N িট িজিনস আেছ (*N ≤* 20). এেদর িতিটর কনার স াব তা দওয়া আেছ *p*1*,p*2 *...pN*. তামােক বলাআেছ য মাট rিটিজিনস কনা হেয়েছ। তাহেল িতিট িজিনেসর জ বলেত হেব য সিট কনার স াব তা কত।

**সমাধান:** মেন কের বস না য i তমিট কনার স াব তা তা দয়াই আেছ *pi*. এখােন িক অিতির আেরকিট কথা বলা আেছ। "তুিম যিদ জােনা য r িট িজিনস কনা হেয়েছ" তাহেল কান িজিনস কনার স াব তা কত তা বল। তামােক যখন *p*1*,p*2 *...pN* ইত ািদ বলা িছল সটা িছল unconditional. কােনা শত নই, কােনা ধরা বাধা নই িকছু নই। এই অব ায় i তম িজিনস কনার স াবনা হেলা *pi*. িক তামােক যা িজ াসা কেরেছ তােত িক একটা বাড়িত condition যাগ কের িদেয়েছ- যিদ তুিম জােনা য r িট িজিনস কনা হেয়েছ তাহেল বল ...। ব াপারটা একটা ছাট উদাহরেণ বুঝােনা যাক। মেন কর Real Madrid আর Barcelona এর মােঝ ফাইনাল খলা হে । য িজতেব স একটা cup পােব। এখন খলার আেগ তামার ফসবুেকর িবেশষ (!) ব ুেদর status পেড় তুিম বুঝেত পারছ য যেহতু আজ Messi খলেছ না তাই Barcelona জতার স াবনা 30% আর Real Madrid এর 70%. িক মেন কর পরিদন তুিম ফসবুক খুেল দখেল য Barcelona কাপিজেতেছ। এখন যিদ তামােক কউ িজ াসা কের "Barcelona কাপ িজেতেছ, এখন বল barcelona খলায় িজেতেছ তার স াবনা কত"। এর উ র সহজ তাই না? "100%". কারণ তামােক বলা হেয়েছ "Barcelona িজেতেছ" এই সােপে বল য Barcelona জতার স াবনা কত। এই সম ােতও একই ঘটনা ঘটেছ। তামােক বলা হেয়েছ য "r িট িজিনস কনা হেয়েছ" তাহেল বল কান িজিনস

কনার স াবনা কত।

ইটা িজিনস নয়া যাক। ধরা যাক *p*1 = 0*.*3 আর *p*2 = 0*.*8. তাহেল-

X 1 ও 2 িটই কনা হেয়েছ তার স াবনা 0*.*3 *×* 0*.*8=0*.*24

X 1 কনা হেয়েছ িক 2 কনা হয় নাই তার স াবনা 0*.*3 *×* 0*.*2=0*.*06

X 1 কনা হয় নাই িক 2 কনা হেয়েছ তার স াবনা 0*.*7 *×* 0*.*8=0*.*56

X িটই কনা হয় নাই তার স াবনা 0*.*7 *×* 0*.*2=0*.*14

এখন মেন কর তামােক বলা হেলা য r = 1 িট িজিনস কনা হেয়েছ। এর মােন মােঝর ইিট ঘটনার কােনা একিট ঘেটেছ। আমােদর মাট ঘটনার space চারিট থেক কেম িট ত নেম গেছ। আেগ চারিটর যাগফল িছল 1 এখন মােঝর িটর যাগফল 0*.*06+0*.*56 = 0*.*62 এিট হল আমােদর ঘটনার space. এখন এই space এ ি তীয়িট (অথ াৎ 1 কনা হেয়েছ িক 2 কনা হয় নাই) ঘটার স াবনা 0*.*06/0*.*62 আর তৃতীয়িট ঘটার স াবনা (অথ াৎ 1 কনা হয় নাই িক 2 কনা হেয়েছ) 0*.*56/0*.*62. তাহেল 1 কনার স াবনা 6/62 আর 2 কনার স াবনা 56/62. বুঝেত পারছ তা কীভােব িহসাব করলাম? থেম বর করলাম আমােদর দওয়া condition ঘটার স াবনা কত। সটাই হেলা আমােদর এখনকার space. এবার এই space এর িতিট ঘটনা ঘটার স াবনা বর কেরিছ। চাইেল িজিনসটােক িকছুটা weighted িহসাব িনকাশ মেন করেত পােরা। আমােদর িট ঘটনা ঘটার স াবনা সমান না, এরা হেলা 0.56 আর 0.06। তরাং তােদর িতিট ঘটার স াবনা হেলা 0*.*56 0*.*56+0*.*06 আর 0*.*06 0*.*56+0*.*06.

এখন তা বলেব- r এর মান 1 বেলই এেতা সহেজ হেয়েছ। n = 4 আর r = 2 এর জ কের দখাও না, মাথার ঘাম পােয় পড়েব! াঁ হয়েতা হােত হােত পুেরাটা করেত অেনক ক হেব িক তুিম যিদ মূল আইিডয়াটা বুেঝ থােকা তাহেল আর পুেরাটা হােত হােত করার দরকার নই। মেন কর তুিম r িট িজিনস যত ভােব কনা যায় তার িতিটেক িল করেল এবং তােদর স াব তা বর করেল। তাহেল আমােদর space হেব এই সব স াব তা িলর যাগফল (S)। আর যিদ িজ াসা করা হয়

৪৬

1 কনার স াবনা কত? তাহেল এই িলে র দেখা কান কানটায় 1 আেছ। তােদর স াব তার হেত যাগফল পাের বর সেহতু কের (তােক *Nr*

S িদেয় ভাগ করেলই হেয় যােব। তাই না? যেহতু N এর মান মা 20 ) আসেল খুব একটা বড় না। তামরা চাইেল backtrack কের N িট িজিনস থেক r িজিনস িনব াচন করেত পােরা। অথবা চাইেল bitmask এর লুপ চািলেয়ও করেত পােরা। এই ব াপাের আমরা থম অধ ােয়ই আেলাচনা কেরিছ। তরাং এখােন এই িনেয় বলার দরকার নই।

তরাং তামরা তামােদর পছে র মথড ব বহার কের এই সম া সমাধান কের ফল।

UVa 580 Critical Mass

**সমস া:** তামােক N দওয়া থাকেব (*N ≤* 30). বলেত হেব কত িল N দেঘ র শ আেছ যা U আর L িদেয় তির এবং শ িটর কাথাও না কাথাও পাশাপািশ 3 িট U আেছ।

**সমাধান:** কাথাও পাশাপািশ িতনিট U আেছ বর করার চেয় পাশাপািশ কাথাও িতনিট U নই বর করা সহজ। কন? কারণ পাশাপািশ িতনিট U " কােনা না কােনা জায়গায়" থাকেত হেব এটা িহসােবর থেক " কাথাও" থাকেব না িহসাব করা সহজ। তুিম মেন কর এেক এেক character িল বসা । পাশাপািশ িতনিট U আেছ িকনা এটা জানার জ তামার আেগর িট character তা জানেত হেবই সই সােথ জানেত হেব ইতমেধ এর আেগ কাথাও পাশাপািশ িতনিট U িছল িকনা। িক যিদ আমােদর সম া হয় পাশাপািশ িতনিট U বসেব না, তাহেল ধু আেগর িট character জানেলই চলেব, কারণ আমরা কখনই পাশাপািশ িতনিট U বসাব না। তরাং আমরা যিদ বর করেত পাির য N দেঘ র কত িল শে পাশাপািশ িতনিট U নই তােক আমরা যিদ 2*N* হেত বাদ দই তাহেলই আমােদর উ র পেয় যাব।

এখন হেল কীভােব বর করেব কত িল শ আেছ যােত পাশাপািশ িতনিট U নই। এিট একিট counting সম া এবং counting সম ায় DP হর হােমশায় ব বহার হেয় থােক। মেন কর *dp*[*n*] হল n দেঘ র valid শে র সংখ া। এখন একিট n দেঘ র valid শ িতন উপােয় তির হেত পাের-X েত L. সে ে আমরা বািক *n −* 1 দঘ *dp*[*n −* 1] ভােব বানােত পাির।

X েত UL. সে ে আমরা বািক *n −* 2 দঘ *dp*[*n −* 2] ভােব বানােত পাির।

X েত UUL. সে ে আমরা বািক *n −* 3 দঘ *dp*[*n −* 3] ভােব বানােত পাির।

আমরা িক েত UUU বসােত পারব না। তরাং আমােদর recurrence হেব *dp*[*n*] = *dp*[*n −* 1] + *dp*[*n −* 2] + *dp*[*n −* 3]. Base case আশা কির িনেজরা বর কের িনেত পারেব। Base case আসেল তমন িকছু না। কান কান n এর জ আমরা আমােদর ফমু লা ব বহার করেত পারব না তাই base case. খয়াল কর য *n* এর মান যিদ 1, 2 বা 3 হয় তাহেল আমরা আমােদর ফমু লায় negative সংখ া বা 0 এর জ dp জানেত চাি । িক আমরা তােদর মান জািন না (হয়েতা 0 এর জ চাইেল বর করা যায়)। তরাং 1, 2 এবং 3 ক base case মেন করেত পার, এবং তােদর মান কাগেজ কলেম বর কের ফল। এরপর এই ফমু লা ব বহার কের বািক n এর জ *dp*[*n*]

বর কের ফল।

UVa 12034 Race

**সমস া:** একিট িতেযািগতায় n িট ঘাড়া অংশ িনে (*N ≤* 1000). বলেত হেব কতভােব ঘাড়- দৗেড়র ফলাফল হেব। এই সম ার ে একািধক ঘাড়া একই ান দখল করেত পাের। একিট উদাহরণ িদেয় ব াপারটা বুঝা যাক। মেন কর িট ঘাড়া অংশ িনে । মাট িতন রকেমর ফলাফল হেত পাের। এক- িট ঘাড়াই থম, ই- থম ঘাড়া থম আর ি তীয় ঘাড়া ি তীয়, িতন- থম ঘাড়া ি তীয় আর ি তীয় ঘাড়া থম। যেহতু উ র অেনক বড় হেত পাের তামােক উ র mod10056 এ িদেত হেব।

৪৭

**সমাধান:** বুঝাই যাে counting dp. ধরা যাক *dp*[*n*] হেলা n িট ঘাড়ার িতেযািগতায় কত ভােব ফলাফল i িট ঘাড়া হেত আবার পাের। (*ni*এখন একিট i এর লুপ চালাও য এই িতেযািগতায় কত জন থম হেয়েছ। এই ) ভােব িনব াচন করা যায়। এখন বািক *n−i* িট ঘাড়ার িতেযািগতার ফলাফল কত রকম হেত পাের? খুব সহজ *dp*[*n−i*] রকম। তাহেল আমরা যা করব তাহেলা n এর একিট লুপ চালােবা 1 হেত 1000 থম হেয়েছ তার জ । পয এর পর *dp*[*n*] এেক বর এেক করার (*ni*জ । এর ভতের i এর একিট লুপ চালােবা কত জন এখােন আমােদর (*ni*)*dp*[*n −i*] যাগ করেলই আমরা *dp*[*n*] পেয় যাব। ) জানার দরকার। আসেল েতই সব (*ij*(*n*) বর কের রাখেত পােরা। এজ *i*) = (*n−*1

*i−*1)+ (*n−*1

*i*

) এই recurrance ব বহার করেত পােরা। তাহেল O(*n*2) সমেয় সব িকছু generate হেয় যােব।

UVa 1638 Pole Arrangement

**সমস া:** n িট building আেছ (*n ≤* 20) যােদর উ তা যথা েম 1*,*2*,...n*. এই building িল যেকােনা order এ থাকেত পাের। যমন- 4, 2, 3, 1 একিট উদাহরন (িচ ৪.২)। এখন তুিম যিদ বাম িদক থেক দেখা তাহেল িক ধু 4 উ তার building দখেত পারেব। যিদ ডান িদক থেক দেখা তাহেল িতনিট building দখেত পারেব যােদর উ তা 1, 3, 4. তুিম িক 2 উ তার building দখেত পারেব না কারণ সিট 3 উ তার building িদেয় ঢেক গেছ। এখন তামােক বলা আেছ বাম িদক হেত কয়িট building দখা যায় (L) আর ডান িদক হেত কয়িট দখা যায় (R). বলেত হেব n িট building ক কত ভােব সাজান যায়।

নকশা ৪.২: UVa 1638

**সমাধান:** আরও একিট coutning dp সম া। হয়েতা তুিম িচ া করেব য কান order এ building িল বসােব। তুিম থেম থম ােন building বসােব, এর পর ি তীয় ােন। িক তা করেল কান কান building বিসেয়ছ এবং কান order এ বিসেয়ছ তা মেন রাখেত হয়। যেহতু *n ≤* 20 তাই কান order এ কান কান building বিসেয়ছ তা মেন রাখা খুব বিশ হেয় যায়।

আেরকিট আইিডয়া হেত পাের য- building িলর height এর order এ বসান। হয়েতা ভাবেত পােরা 1 হেত i building িল বসােনার পর আমরা যিদ বাম থেক কয়িট building দখা যায় আর ডান থেক কয়িট দখা যায় এই ইিট সংখ া state এ রািখ তাহেলই হেব। িক আমার মেন হয় তােতও হেব না। কারণ যসব building দখা যায় না তােদর অব ানও একিট পূণ ব াপার। মেন কর 20, 10, 30 আর 20, 30. এখন মেন কর 20 আর 30 এর মােঝ 40 বসােব। থম ে

সই বসােনার জায়গা িক িট, িক ি তীয় ে একিট। তরাং দখা যায় না এরকম building এরও িক ভুিমকা আেছ। তাহেল িক করা যায়?

একটু যিদ িচ া কর তাহেল বুঝেব য বাম বা ডান িদক হেত সবেচেয় দূের বা সবেচেয় বড় যই building দখা যােব সটা হেলা n. তাহেল আমরা আমােদর সম ােক একটু সাজা করেত পাির আর তাহেলা বাম হেত n পয L িট আর ডান হেত n পয R িট building দখা যায় কত ভােব। আেরকটা িজিনস খয়াল কর n এর বাম িদেক কয়িট building আেছ সিটই আসল ব পার কান কান

৪৮

height এর আেছ তা িক ব াপার না। মেন কর n এর বােম x িট building আেছ। আমরা চাইেল সসব building এর height ক 1 হেত x এ map করেত পাির। অথ াৎ সবেচেয় ছাট building 1, এর পেরর বড়িট 2 এরকম কের সবেচেয় বড়িট x. তাহেল আমরা আসেল আমােদর সম ােক ইিট সহজ সম ায় ভাগ কের ফলিছ। মেন হয় এটু কথা এখেনা অেনেকর কােছ অপির ার লাগেছ। পির ার কের বলা যাক।

আমরা একিট লুপ চািলেয় িঠক করব n এর বােম মাট কয়িট building আেছ। ধরা যাক i িট। আমরা িকছু পরই আেছ n উ ণ আেগ বেলিছ তার building. এই i তাই িট আমরা building (*n−*1

*i*

এর height কােনা ব াপার না, কারণ এেদর ) ভােব এই i িট building িনব াচন করেত পাির। আমরা চাইেল এই i িট building ক 1 হেত i height এ map করেত পাির। আর বািক *n −* 1*− i* িট building ক ডােন রাখেত পাির আর তােদর একই ভােব mapping করেত পাির (1 হেত *n −* 1 *− i* পয )। তাহেল আমােদর সম া দাঁড়ায় *i* িট building ক কতভােব সাজান যায় যন বাম হেত দখেল *L −* 1 িট building দখা যায় (L তমিট হেব n height এর)। একই ভােব ডান িদেক *n −* 1 *− i* িট building ক কত ভােব সাজান যায় যন ডান হেত দখেল *R −* 1 িট building দখা যায়। আমরা এই িট count ক ন কের i এর জ উ র পােবা। এভােব i এর লুেপর িত ে র জ উ র যাগ করেল আমরা আমােদর আসল উ র পেয় যাব।

তাহেল আমােদর সম া reduce হেয় দাঁড়াল 1 হেত n পয n িট building ক (তােদর height 1 হেত n) কত ভােব সাজােল বাম হেত দখেল X িট building দখা যােব। এটার সমাধান িক ই িদক হেত দখার মত কেরই হেব। থেম n তম building ক বসাও। আর আেরকিট লুপ চালাও এর বােম কয়িট আর ডােন কয়িট building (ncr করেত হেব)। ধরা যাক বােম i িট building. তাহেল তামােক বর করেত হেব i িট building ব বহার কের X - 1 িট building দখা যােব এভােব কত ভােব building িল সাজান যােব। এর মােন আমােদর dp এর state হেব (*n, X*). এর ভতের চলেব i এর একিট লুপ। এবং সিট (*i, X −* 1) ক call করেব। আর বািক *n − i* িট building ক ডান িদেক তা ই া খুিশ মত বসান যােব কারণ আমরা তা তােদর বাম িদক হেত দখেতই পারব না।UVa 1639 Candy

**সমস া:** িট বাে র িতিটেত n িট কের চকেলট আেছ (*n ≤* 200*,*000). তামার ব ু p স াব তায় থম বা িনব াচন কের, তরাং 1 *− p* স াব তায় ি তীয় বা িনব াচন কের। স য বা িনব াচন কের সখান থেক একিট চকেলট খায়। একিদন হেলা িক স তার িনব ািচত বা খুেল দেখ সখােন

কােনা চকেলট নই। বলেত হেব ওপর বাে চকেলেটর expected পিরমাণ কত।

**সমাধান:** আমরা একিট লুপ চালােবা অ বাে কয়িট চকেলট বািক থাকেব তার উপর। ধরা যাক i পিরমাণ বািক থােক। এর মােন এমন একিট পয ায় আসেব যখন একিট বা (a) থেক n িট চকেলট খাওয়া হেয় যােব, ওপর বা (b) হেত *n − i* িট খাওয়া হেয় যােব এবং এই সময় এেস আবার *n* িট চকেলট খাওয়া a বা হেত খাইেত চাইেব।তার মােন এই পয ােয় এেস a বা যােব এবং b বা থেক *n−i* িট খাওয়া হেয় যােব। এই খাওয়ার order মাট থেক (*n*+*n−i*

*n* n িট ) খাওয়া ভােব হেত হেয়

পাের। এখন ধু খাওয়ার order কত রকম হেব সটা বর করেলই তা হেব না, তােদর probability ও বর করেত হেব। a বা হেত n িট খাওয়ার probability হেলা *pn*, b বা হেত *n−i* িট খাওয়ার probability হেলা (1*−p*)*n−i*. একিট *p*. তাহেল মাট probability সই হেব সােথ (2*n−i*

সবেশেষও *n*

)*pn*+1আবার *×*(1*−p*)a বা (*n−i*)হেত . খেট চাইেব, অথ াৎ আরও একই ভােব b বা হেত *n* িট খাওয়ার ঘটনা আলাদা n এর িবশাল মােনর জ ভােব (*ni*) এর িহসাব মান করেত িক হেব। এখন কথা হেলা এই সংখ া কীভােব অেনক বড় হেব, আবার *pn* এর মানও িক বর করবা? অেনক ছাট হেব। তরাং সরাসির িহসাব করেল মেন হয় উ র সিঠক আসেব না। এই ে logarithm খাটােত হয়। আমরা আমােদর সূ সরাসির না ব বহার কের যা করব তা হল তার logarithm বর করব- log(2*n − i*)! *−* log*n*! *−* log (*n − i*)! + (*n* + 1)log*p* + (*n − i*)log(1 *− p*). আর এই িহসােবর পর তার exponent নব (e এর power). এভােব আমরা বর করিছ কত probability ত b বাে (ি তীয় বাে ) i থাকেব। এভােব সব i এর জ probability বর কের তােক i িদেয় ন

৪৯

কের এেদর সবাইেক যাগ করেল আমরা expectation পেয় যাব। একটা িজিনস, log*n*! এর মান কীভােব বর করবা? সহজ, log*n*! = log1 *×* 2 *× ... × n* = log1+log2+*...*+log *n*.

তরাং আমরা েতই সকল দরকাির n এর জ log*n*! এর মান বর কের রাখেত পাির।

UVa 10288 Coupons

**সমস া:** মাট n ধরেনর পন আেছ (*n ≤* 33) আর অেনক িল বা আেছ। তুিম যিদ একটা বা খুল তাহেল তার মােঝ একিট random পন পােব। পনিট n ধরেনর মােঝ যেকােনা ধরেনর হেত পাের। তামােক বলেত হেব expected কত িল বা খুলেল তুিম সব ধরেনর পন পােব।

**সমাধান:** আমার অ তমপছে র expectedvalueএর সম া।কারণ সাধারণত আমরা expectation সম ােক approach কির ∑*iP*(*i*) এর মত কের। িক এই সম ািট সমাধান করেত হয় একটু অ ভােব। মেন কর আমােদর কােছ i িট িভ ধরেনর পন আেছ। তাহেল expected কয়িট বা খুলেল আমরা নতুন ধরেনর পন পােবা? যেহতু আমােদর কােছ ইেতামেধ ই i ধরেনর পন আেছ এর মােন আমােদর কােছ *n − i* ধরেনর পন নই। একিট বা খুলেল এই *n − i* ধরেনর কােনা একিট হবার স াবনা *p* = *n−i n* . তাহেল আমােদর expected 1/*p* িট বা খুলেত হেব নতুন পন পাবার জ । এভােব আমরা *i*+1 িট িভ ধরেনর পন পাবার জ expected বা খুলার পিরমাণ

পেয় যাব। আর এেক এেক আমরা *i* + 2*,i* + 3*...n* পয উ র বর কের ফলেত পারব।

অেনেক হয়েতা ভাবেত পােরা নতুন পন পাবার probability *p* হেল expected বা খুলার পিরমাণ 1/*p* কন? এটা খুব একটা কিঠন না। নতুন পন পেত j িট বা খুলার দরকার মােন থম *j−*1 বাের নতুন পন উেঠ নাই, যার স াবনা (1*−p*)(*j−*1) আর j তম বাের নতুন পন পাওয়া গেছ, যার স াবনা *p*. তরাং j িট বা খুলার পর নতুন পন পাবার স াবনা (এবং j িট বা খুলার আেগ িক নতুন পন পাবা না) *P*(*j*) = *p ×* (1 *− p*)(*j−*1). তরাং আমােদর expected value হেব ∑*j × P*(*j*) = ∑*j × p ×* (1 *− p*)(*j−*1). তামরা চাইেল এই series sum বর করেত পােরা অথবা একটু calculus খাটােত পােরা। আমরা জািন 1/(1 *− p*) = ∑*pj*. আমােদর সমীকরণটা ায় একই ধরেনর। থমত আমােদর মূল সমীকরেণ (1 *− p*) এর power নওয়া হি ল। আমরা আমােদর series sum এ p এর বদেল 1*−p* বসাই। 1/*p* = ∑(1 *− p*)*j*. এই সমীকরণেক তুিম যিদ p এর সােপে differentiate কর তাহেল পাবা *−*1/*p*2 = ∑*−j ×* (1 *− p*)(*j−*1). আর একটু এিদক অিদক করেলই 1/*p* = ∑*j × p ×* (1 *− p*)(*i−*1) পেয় যাবা। তেব এটা মেন হয় একটু কিঠন হেয় গল। আেরকটা উপােয় দখােনা যাক। মেন কর আমােদর expected value হেব E. এখন p স াব তায় থম বােরই আমরা নতুন পন পেত পাির 1*×p*। অথবা 1*−p* স াব তায় থম বার নতুন পন উেঠ নায়। তাহেল এর পের নতুন পন পেত তামােক E িট বা খুলেত হেব (এখন য একিট খুলেল সিট বােদ) অথ াৎ (1 *− p*)(1 + *E*)। তরাং আমােদর সূ দাঁড়াল *E* = *p* + (1 *− p*)(1 + *E*) আর এটােক সমাধান করেল দাঁড়ােব *E* = 1/*p*.

UVa 11346 Probability

**সমস া:** (*−a,−b*) হেত (*a, b*) পয rectangle িটএকিট special জায়গা। তামােক এই special জায়গার মােঝ একিট (*x, y*) িব randomly িনব াচন করেত হেব (x আর y ক য পূণ সংখ া হেত হেব এমন কথা নই) যন (0*,*0) ও (*x, y*) িব িদেয় তির rectangle িটর ফল S এর হেত বড় হয়। S, a, b এর মান তামােক ইনপুট িহসােব দওয়া থাকেব। বলেত হেব সরকম একিট িব িনব াচেনর স াবনা কত।

**সমাধান:** যারা ইেতামেধ ই কেলেজ calculus পেড়ছ তারা জােনা য এসব সম ার ে আম- রা একিট quadrant এ িহসাব িনকাশ কির কারণ অ সব quadrant এ সব িহসাব িনকাশ symmetric হয়। এখােন যমন আমােদর special জায়গা এবং (*x, y*) এর সােথ মূল িব র rectangle সবই symmetric. তরাং আমরা ধু first quadrant এ িহসাব িনকাশ করব।

৫০

তাহেল আমরা ধু (0*,*0) হেত (*a, b*) এইটু special জায়গা িনেয় মাথা ব াথা করব। কথা হল এর মােঝ কান টু জায়গা থেক (*x, y*) িনব াচন করেল মূলিব র সােথ ফল S এর বড় হেব। একটু িচ া কের দেখা, *xy* = *S* রখার িব িলর সােথ মূল িব িঠক িঠক S ফল তির কের। এর উপেরর (up) িব িলর সােথ S এর থেক বিশ এবং এর িনেচর িব িলর সােথ S এর থেক কম ফেলর আয়তে তির কের। তাহেল আমােদর বর করেত হেব *xy* = *S* এই রখার িনেচ special জায়গার ফল। এই পিরমান যিদ আমরা *ab* হেত িবেয়াগ কির তাহেল আমরা পাব সই অংেশর ফল যা হেত (*x, y*) িনব াচন করেল আয়তে ে র ফল S হেত বড় হয়। এই ফলেক যিদ আমরা *ab* িদেয় ভাগ কির তাহেল আমরা আমােদর উ র পেয় যাব। এখন কথা হল এই *xy* = *S* রখার িনেচর অংেশর ফল কীভােব বর করব।

4

3(4*,*3)

2*xy* = 4

100 1 2 3 4 5 নকশা ৪.৩: (*a, b*) = (4*,*3) এবং *S* = 4

আমরা একটা উদাহরণ িনেয় কাজ কির তাহেল মেন হয় িবধা হেব। ধরা যাক (*a, b*) = (4*,*3) আর *S* = 4. তাহেল first quadrant এর যই িচ হয় তা িচ ৪.৩ এ দওয়া হেলা। *xy* = *S* = 4 এিট একিট বাঁকা রখা দখেতই পা । আর সই সােথ আমরা (0*,*0) হেত (4*,*3) এই আয়তে ও এঁেকিছ। আমােদরেক িচে র shaded জায়গার ফল বর করেত হেব। কারণ এই জায়গােত যিদ আমরা কােনা (*x, y*) িনব াচন কির তাহেল মূল িব র সােপে তার আয়তে ে র ফল S অেপ া কম হেব। দখেলই বুঝেত পারছ shaded জায়গােক ই অংেশ ভাগ করেত পােরা। একটা হেলা ছাট আয়তে আেরকটা হেলা *xy* = 4 এর িনেচর অংশ। ছাট আয়তে ে র উপেরর ডান কাণা আেছ (*S*/*b, b*) = (4/3*,*3) ত। যার ফল *S*/*b × b* = 4/3 *×* 3=4= *S*. বািক থােক integration

*S*/*b* = ∫4/3 *aS*/*b*

*Sxdx* হেত = *a S* = ln*x*4 ∣∣∣পয *aS*/*b* = *xy S*(ln(*a*)*−*ln(*S*/*b*)) = 4 এর িনেচর অংেশর ফল। এিট = *S* ln( *abS* )১. একিট সহজ তামরা চাইেল

a বা b এর মান বিসেয় এই মান বর করেত পােরা। তাহেল আমরা অ অংেশর ফল বর করলাম যখােনর িব িলর সােথ মূলিব র আয়তে ে র ফল S এর কম। আমরা যিদ *ab* হেত এই ফল বাদ দই তাহেল (0*,*0) হেত (*a, b*) পয সই অংেশর ফল পােবা যখােনর িব িলর সােথ মূলিব র আয়তে ে র ফল S এর বিশ। এবং সবেশেষ এেক যিদ আমরা *ab* িদেয় ভাগ কির তাহেল সই িব িনব াচেনর probability পেয় যাব। আর একটু যিদ সাবধান থাক তাহেল একটা special case াে ডল য করেত হেব তা ইতমেধ ই বুেঝ গছ। Case টা হেলা যিদ (0*,b*) হেত (*a, b*) এই লাইন *xy* = *S* ক ছদ না কের, অথ াৎ *S*/*b>b*.

১ln (*a*) *−* ln (*b*) = ln (*a*/*b*)

৫১

UVa 10900 So you want to be a 2*n*-aire?

**সমস া:** একিট game show ত েত তামােক 1 টাকা দওয়া হেব। এরপর এেক এেক তুিম n রাউ ড খলেত পারেব (*n ≤* 30)। িতিট রাউে ডর েতই তামােক একিট করা হেব। পারার স াবনা t হেত 1 এর মােঝ কােনা একিট random সংখ া (0 *≤ t ≤* 1). তুিম চাইেল িটর উ র না িদেয়ই quit করেত পােরা। quit করেল তামার কােছ যত টাকা আেছ তা তামার হেয় যােব। আর যিদ quit না কর তামােক ে র উ র িদেত হেব। তুিম যিদ ে র উ র িদেত পােরা তাহেল তামার টাকার পিরমাণ ি ণ হেয় যােব এবং তুিম পেরর রাউে ড যােব। আর যিদ ভুল হয় তাহেল তামার টাকার পিরমাণ 0 হেয় যােব এবং খলা শষ হেয় যােব। তুিম তামার expected টাকার পিরমাণ maximize করেত চাও। বলেত হেব এই expected টাকার পিরমাণ কত।

**সমাধান:** আমরা dp এর মত করব। আমরা বর করব যিদ আমরা i তম রাউে ড থািক তাহেল expected টাকার পিরমাণ কত। আমােদর base case িক হেব? মেন কর আমরা n তম রাউে ড আিছ, তখন যিদ আমরা িঠক ে র উ র িদতাম তাহেল *n*+1 রাউে ড উঠতাম। এখােন আর কােনা নই। এখােন আমােদর খলা শষ করেতই হেব আর তাহেল আমরা 2*n* টাকা পােবা। তরাং আমােদর base case হেব *f*(*n*+1) = 2*n*. এখন recurrence এর পালা। হল i তম রাউে ড expected টাকার পিরমাণ কত। আমরা জািন আমােদর ে র উ র পারার স াবনা t হেত 1 এর মােঝ যেকােনা একিট সংখ া। ধরা যাক p. তামার হােত ইিট অপশন। এক- আমরা quit করব তাহেল আমরা 2*i−*1 টাকা পােবা। ই- ে র উ র িদব। যিদ ে র উ র দাও তাহেল কত টাকা পাবা? p স াব তায় *f*(*i*+1) টাকা পাবা আর (1-p) স াব তায় 0 টাকা। তাহেল ে র উ র িদেল expected টাকার পিরমাণ হেব *p×f*(*i*+1). এখন তুিম চাও তামার টাকার পিরমাণ maximize করেত। তাহেল িক করেব? ে র উ র দেব নািক দেব না? সহজ উ র- যা করেল expected টাকার পিরমাণ বিশ হেব সটা করবা। অথ াৎ *max*(2*i−*1*,p × f*(*i* + 1)). এিট হেলা p এর জ expected টাকার পিরমাণ। িক p এর মান তা t হেত 1 পয হেত পাের। তাহেল? তাহেল যা করেত হেব তাহেলা তুিম t হেত 1 পয সব মােনর জ expected টাকার পিরমাণ যাগ কর এবং সই সােথ t হেত 1 পয যত িল সংখ া আেছ তা িদেয় ভাগ কর। িক t হেত 1 পয তা অসংখ সংখ া আেছ। িক করবা? উপায় হেলা integration- 11*−t*

∫ *t* 1*max*(2*i−*1*, pf*(*i* + 1))*dp*. কন (1 *− t*) িদেয় ভাগ হেলা? কারণ তুিম t হেত 1 পয integrate কেরছ। এটা িকছুটা গড় নবার মত।এখন কথা হেলা এই integration করবা কীভােব? *f*(*i* + 1) একিট constant কারণ এিট p এর সহজ তাহেল উপর কের আমােদর িনভ িলখেত র কের max পাির না। এর ধের 1*−t*

1মান ∫ নই *t* 1হেব *max*(2*C* 2= *i−*1 *f*(*i*+1). *i−*1আর *, pC*)*dp*. নাহেল তাহেল হেব যিদ আমরা *pC*. *pC* তাহেল আমােদর *<* integration ক একটু 2*i−*1 *⇒ p <* 2*i−*1/*C* হয় আমরা আমােদর integral ক ই অংেশ ভাগ করেত পাির t হেত 2*i−*1/*C* আর 2*i−*1/*C* হেত 1. ধু একটা িজিনস খয়াল রেখা য 2*i−*1/*C* যিদ t এর থেক ছাট হয় বা 1 এর থেক বড় হয় তাহেল যন তুিম উলটা পালটা িকছু না কর।

তাহেল এই সম ার মাধ েম তামরা দখেল কীভােব expectation এর সম ােত আমরা calculus ব বহার করেত পাির। আর Integration এর ফাংশন যিদ max ওয়ালা হয় তাহেল আমরা কীভােব তােক ভে িলেখ সমাধান করেত পাির।

UVa 11971 Polygon

**সমস া:** তামােক একিট N দেঘ র একিট সরলেরখা দওয়া আেছ। তুিম এেক randomly K িট জায়গায় কাটেল (1 *≤ N ≤* 106*,*1 *≤ K ≤* 50)। এর ফেল য *K* + 1 িট সগেম ট তির হেলা তা িদেয় একিট polygon গঠন করেত পারার probability কত?

**সমাধান:** িনঃসে েহ এিট একিট কিঠন সম া। সিত কথা বলেত এিট পড়ার সােথ সােথ আিম কােনা idea পাই নাই। যিদ pattern খাঁজার মাধ েম সমাধান করেত চাও তাহেল বশ সহেজই

৫২

সমাধান হেয় যােব। তেব যিদ িনেজ থেক গািণিতক ভােব সমাধান করেত চাও তাহেল েসসটা খুব একটা সহজ হেব না। নেট খুঁেজ chinese site হেত আিম অ ত 4 িট িভ ধরেনর সমাধান দেখিছ। এ থেক িট এখােন দখােনা যাক।

যভােবই সমাধান কর না কন মূল আইিডয়া হেলা *K* + 1 িট সগেমে টর মােঝ সবেচেয় বড় সগেমে টর দঘ 1/2 এর থেক কম হেব। অথ াৎ আমােদর বর করেত হেব িতিট সগেমে টর দঘ 1/2 এর থেক ছাট হবার probability কত? কন 1/2 এর থেক ছাট হেব? মেন কর সবেচেয় বড় সগেম টটা িনলা। এরপর বািক সগেম ট িল জাড়া লািগেয় লািগেয় তুিম বড় সগেমে টর অ মাথায় পউছাবা। যিদ বড়টার দঘ 1/2 এর থেক বড় হয় তাহেল তুিম কখনই অ াে পাঁছােত পারেব না। আর যিদ 1/2 এর সমান হয় তাহেল কবল অপর াে পাঁছােব একিট লাইন তির কের, polygon হেব না। আর যিদ সবেচেয় বড়টা 1/2 এর থেক ছাট হয় তাহেল আমরা polygon বানােত অব ই পারব। ও াঁ ভাল কথা, বেলেম বেলিছ সগেমে টর দঘ N িক এখােন আিম এই দঘ 1 ধের কাজ কেরিছ। কন? কারণ, আমােদর উ র আসেল N এর উপর িনভ র করেব না তাই না? একিট 1 দেঘ র যখােন যখােন কাটা valid, 100 দেঘ র সখােন সখােন কাটা valid। মােন propotionally কাটেত হেব- এই আর িক। যিদ তুিম 1 এর 1/3 ত কাট তাহেল 100 এর 100/3 ত কাটবা- এরকম। যিদ কাটা িল 1 এর ে valid হয় তাহেল 100 এর ে ও valid হেব। তরাং আমােদর N এর দরকার নই। আমরা আমােদর িহসাবেক সহজ করার জ ধের িনেত পাির সগেমে টর দঘ 1. এখন আমােদর থম সমাধান দখা যাক।

আমরা " িতিট segment এর দঘ 1/2 এর থেক ছাট হেব" এিটর probability না বর কের " কােনা একিট segment এর দঘ 1/2 এর সমান বা বড় হেব" এটার probability বর করব। খয়াল কর কােনা একিট segment যিদ 1/2 এর সমান বা বড় হয় তাহেল আর কউ িক 1/2 এর সমান বা বড় হেব না কারণ বািক যই অংশ আেছ তােত অ ত িট সগেম ট বসােত হেব (আিম ধের িনি *K >* 1 কারণ *K* = 1 হেল িট সগেম ট হয় আর িট বা িদেয় কখনই polygon হয় না)। মেন কর আমরা K বার randomly কেটিছ এবং *K* + 1 িট সগেম ট পেয়িছ। ধরা যাক এেদর দঘ হেলা *x*0*,x*1*,...xk* এবং *x*0 হল সবেচেয় বড়টা। আমরা এেক সরলেরখার একদম বাম সাইেড রাখেবা। তাহেল *x*0 এর অ মাথা অব ই [1/2*,*1] এর মােঝ থাকেব। ধু তাই না, আসেল যই *K* িট িব েত কাটা হেব সব িলই [1/2*,*1] এর রে পড়েব। এই probability কত? কােনা একিট িব [0*,*1] এই রে র শষ অেধ েক পড়েব তার probability 1/2. তাহেল *K* িট িব ই শষ অেধ েক পড়েব তার probability হেব K টা 1/2 ন অথ াৎ 12*K* . তাহেল এটাই িক উ র? না, এখােন আমরা ধের িনেয়িছ য *x*0 অথ াৎ সবেচেয় বড় segment িট বােম থাকেব। িক তুিম যখন random ভােব K জায়গায় কাটেব তখন তা সবেচেয় বড়টা বােম নাও থাকেত পাের। তাহেল উপায় িক? মেন কর তুিম random ভােব K িট জায়গায় কাটার পর segment িলর বাম থেক ডােন দঘ

িল হেলা *y*0*,y*1*,...yK*. তুিম cyclically ঘুরােত থােকা যত ণ না সবেচেয় বড়টা বােম আেস অথ াৎ তুিম দখবা *y*1*,y*2*,...yK,y*0 এরপর *y*2*,...,yK,y*0*,y*1 এরকম কের তত ণ cyclically ঘুরােত থাকবা যত ণ না সবেচেয় বড়টা বােম আেস। তাহেল এভােব তুিম যেকােনা ভােব কাটার জ ঘুিরেয় ঘুিরেয় বড়িটেক সবেচেয় বােম আনেত পারেব। যিদ এটা সিত হয় তাহেল একবার ভাব যিদ েতই সবেচেয় বড়টা বােম থােক তাহেল তােক ঘুিরেয় ঘুিরেয় িন য় িবিভ জায়গায় নওয়া স ব? যেহতু মাট *K*+1 িট সগেম ট আেছ তাই এেক *K* বার ঘুরান যােব আর িনেজ একটা, মাট *K* + 1 অব ায় িনেয় যাওয়া স ব। এর মােন আসেল কােনা একিট সগেম ট 1/2 এর থেক বড় পাবার স াবনা *K*+1

2*K* কারণ বােম সই বড়টা পাবার স াবনা 12*K* আর তােক *K* + 1 ভােব ঘুিরেয় নানা জায়গায় িনেয় যাওয়া যায় আর এই সব নানা জায়গা তামােক সব অব া িদেত সাহায করেব। তাহেল আমােদর polygon বানােত পারার probability হেব 1 *− K*+1

2*K* . এবার ি তীয় সমাধান দখা যাক। আমরা calculus ব বহার করব। ধরা যাক আমােদর সবেচেয় বড় segment িটর দঘ *x*. আমরা x এর সােপে integration করব। যেহতু এিট 1/2 এর থেক বড় হেব তাই আমােদর integration এর র হেব [0*.*5*,*1]. এখন আমােদর বড় segment িট এক সাইেড হেত পাের বা ভতের হেত পাের। ধরা যাক এিট বাম সাইেড। তাহেল বাম সাইড হেত িঠক *x* দূরে একিট কাটা হেব। যিদ আমরা *dx* ক খুব ছাট জায়গা ধির তাহেল এই *x* দূরে একিট কাটা হবার স াবনা *dx*1 = *dx* যখােন 1 হল পুেরা সরলেরখার দঘ । এখন এই কাটিট *K* িট কােটর যেকােনািটই হেত পাের। কততম কাটিট এখােন? সজ একিট *K* ন হেব (কত ভােব

৫৩

ওখােন কাট হেত পাের? K ভােব। কারণ K কােটর যেকােনা কাটই ওখােন হেত পাের)। আর এই কাট ছাড়া বািক কাট িল ডান িদেকর বািক 1*−x* জায়গায় হেব। তরাং বািক *K −*1 কােটর িতিট ডােনর (1 *− x*) এ পড়ার probability (1 *− x*)*K−*1. তরাং বােম x দেঘ র কাট হবার স াবনা *K*(1 *− x*)*K−*1*dx*. একই ভােব যিদ x অংশিট যিদ ডান িদেক হয় তাহেল একই probability ( তরাং 2 িদেয় ন হেব). এখন ধরা যাক আমােদর বড় সগেম টিট সাইেড না হেয় মােঝ। তাহেল এই ে এই সগেম ট এর বােম একিট িব পড়েব আর ডােন আেরকিট। মেন কর আমরা ডান হেত x অংশ অ রেঙ রং কেরিছ। তাহেল আমােদর x এর বাম মাথা আসেল রং না করা যেকােনা জায়গা থেক হেত পাের। রং না করা অংেশর দঘ (1 *− x*) তাই এই অংেশ একিট কাট হবার স াবনা (1*− x*). এখান থেক x দূর ডােন আমােদর আেরকিট কাট হেত হেব। এর স াবনা *dx*. এখন খয়াল কর K িট কােটর মােঝ x এর বাম মাথায় একিট আর ডান মাথায় আেরকিট কাট আেছ। বােমরিট *K* িট কােটর যেকােনািট হেত পাের আর ডােনরিট হেত পাের (*K −* 1) ভােব (উদাহরণ প- তৃতীয় কাট বােম আর থম কাট ডােন)। তাহেল এই িট কাট হেত পাের *K*(*K −*1) ভােব। তরাং এেদর ন করেল x এর ই িদেক িট কাট হবার স াবনা বর হেয় যােব। বািক থােক বািক *K −* 2 িট কাট। এরা আসেল x বােদ বািক জায়গার যেকােনা কাথাও হেত পাের। x বােদ বািক অংেশর দেঘ র যাগফল (1 *− x*) তাই এটা হেত পাের (1 *− x*)*K−*2 ভােব। তরাং x মােঝ হবার স াবনা *K*(*K −* 1)(1 *− x*)(1 *− x*)*K−*2*dx*. তাহেল x মােঝ হবার স াবনা আর সাইেড হবার স াবনােক যাগ কের যিদ integration কর তাহেল উ র পেয় যােব।

েতই বেলিছ িনঃসে েহ এিট একিট কিঠন সম া। এবং একবার পেড়ই তুিম এটা বুঝেত পারবা আিম আশা কির না। সিত কথা বলেত চাইিনজ িলখােক translate কের এর পর এই সমাধান বুঝেত আমার বশ অেনক ণ লেগেছ।

UVa 1640 The Counting Problem

**সমস া:** তামােক িট সংখ া a ও b দওয়া হেব (1 *≤ a ≤ b ≤* 108). বলেত হেব এেদর মােঝ যত িল সংখ া আেছ (এই িট সহ) তােদর সব িলেত কান digit কয়বার কের আেছ। অথ াৎ 0 কয়বার, 1 কয়বার এভােব 9 পয সব digit কয়বার কের আেছ বলেত হেব। যিদও মূল সম ােত *a ≤ b* বলা নাই িক এখােন আেলাচনার িবধার জ আিম এই condition আেরাপ কেরিছ। যিদ *b<a* হয় তাহেল ধু তােদর swap কের িনেলই হেব।

**সমাধান:** এিট একিট র সহজ mathematical সম া। যসব সম ােত বেল য *a* ও *b* এর মােঝ িকছু একটা count করেত বা িহসাব করেত তখন সাধারণত যা করেল সহজ হয় তাহেলা তুিম একিট ফাংশন *f*(*n*) িলখেব যটা 1 হেত *n* পয উ র বর কের। এর পর তুিম *f*(*b*) *− f*(*a −* 1) করবা, তাহেল তুিম a হেত b পয উ র পেয় যােব। আমরা এই সম ােতও এই ি ক খাটােত পাির। তরাং আমােদর বেলম দাঁড়াল, একিট মান n দওয়া হেল তুিম কীভােব 1 হেত n পয সব সংখ ার িডিজট িলর সংখ া বর করেব। বুঝার িবধার জ আমরা একটা উদাহরণ নই- 315097. আমরা 1 হেত 315097 পয সংখ া িলর িডিজেটর সংখ া বর করেত চাই।অথ াৎ এই সংখ া িলেত কান িডিজট কয়িট কের আেছ। এিট একিট 6 অংেকর সংখ া। আমরা এেক এেক িতিট অংেক যাব এবং সখােন 0 হেত 9 পয লুপ চালােবা আর িজ াসা করব এই ােন এই িডিজট 1 হেত n পয কয়বার আসেত পাের। মেন কর আমরা জানেত চাি *i* তম ঘের (ডােনর ঘর অথ াৎ lsb এর জ *i* = 0, এর বােমর ঘেরর জ *i* = 1 এরকম) d িডিজটিট কত বার আেস। এই ে d এর লুপ চলেব 0 হেত 9 পয , আর i এর লুপ চলেব 0 হেত 5 পয (কারণ সংখ ািট 6 িডিজেটর আর আমরা 0 indexing ধেরিছ)। ধরা যাক 315097 এর জ i = 3 ও d = 1 এর জ এই count আমরা বর করেত চাই। িতনিট িজিনস দখেত হেব *i* = 3 এর বােম িক আেছ (L = 31)? ডােন িক আেছ (R = 097)? আর িনেজ িক (S = 5)? থেম দখেত হেব d আর S এর স ক িক। ছাট, বড় না সমান। িতন ে িতন রকমভােব গণনা করেত হেব। *d* = 1 এর জ এিট *S* = 5 হেত ছাট। এই ে ক না কর কখন কখন এখােন 1 আসেব? *i* = 3 এ যখন 1 আসেব তখন আমােদর বােমর সংখ া িক 0 হেত 31 পয মাট 32 (*L* +1) িট সংখ ার যেকােনািট হেত পাের। আবার আমােদর

৫৪

ডােনর সংখ া 000 হেত 999 পয মাট 1000 (10*i*) রকম হেত পাের। তাহেল 1 আসেব মাট 32 *×* 1000 বার বা ফমু লায় িলখেল (*L* + 1)10*i* বার। যিদ *d>S* হয়? তাহেল িক আমােদর বােমর সংখ া 31 হেত পারেব না। কন? ধর *d* = 7 *> S* = 5 এখােন দেখা (31)7(???) এরকম কােনা সংখ া িক 1 হেত 315097 এর মােঝ নই। তার মােন আমােদর বােমর অংশ হেত পাের 0 হেত 30 পয মাট 31 রকম। আর ডােনর সংখ া আেগর মতই 1000 রকম। এ হেত আমরা বর করেত পাির 7 কয় বার আসেব। বািক থােক *d* =5= *S* = 5 এর case. এই ে বােম 31 আসেত পারেব িক যখন 31 আসেব তখন আবার ডােন 000 হেত 999 পয সব িল সংখ াই হয়েতা আসেত পারেব না। কবল 000 হেত 097 (R) পয আসেত পারেব। বােম 0 হেত 30 হেল িক হেব তােতা *d>S* এর case থেকই বর কের ফেলছ। আর যিদ বােম L এর সমান হয় তাহেল ডােন *R* + 1 রকম সংখ া আসেত পাের। তরাং এই ভােব আমরা সকল i এর ােন সকল d এর সংখ া েন ফলেত পারব।

তেব িকছু িজিনেস সাবধান হেত হেব। যমন আমােদর এই উদাহরেণ *i* = 5 এ *d* = 0 কখনই হেব না। আবার যখন *i* = 0 হেব তখন R এর মান কত হেব? আসেল আিম িনেজও জািন না য R এর মান কত ধরেল িমলেব। আিম হেল এখােন কােনা risk নব না। বরং এখােন R ছাড়াই িহসাব করব। ধর 315097 এ শষ ঘের (অথ াৎ *i* = 0 এর ে ) 1 কত বার আেস? সহজ 1 যতবার আসেব এর বােম 0 হেত 31509 পয মাট 31510 িট সংখ া আসেব। আর এর ডােন যেহতু কােনা ঘর নাই তাই R এর িহসাব করার কােনা দরকার নই। একই ভােব *d* = *S* বা *d>S* এ

েসস করেত হেব।

আবােরা বিল, এই সম া বেল বুঝােনা কিঠন। আিম তামােক খুব জাড় সমাধােনর পথ বেল িদেত পারব। িক তামােক িনেজ থেক এটা বুঝেত হেব। কারণ এেত অেনক step আেছ এবং িতিট step এর জ ফমু লা বর করেত হেব। আিম িনি ত এর থেক হয়েতা সহজ উপােয় িহসাব করা স ব যটা আমার মাথায় এই মুহূেত আসেছ না। তামরা কউ সরকম সমাধান পেল সবাইেক জািনেয় িদও।

UVa 10213 How Many Pieces of Land?

**সমস া:** একিট ellipse এর পিরিধর উপর n িট িব নাও (1 *≤ n ≤* 231)। এবার িত িট িব িনেয় তােদর মােঝ রখা টান। এেত কের ellipse িট অেনক ভােগ িবভ হেব। বলেত হেব সেব া কত ভােগ ভাগ করা স ব। **সমাধান:** এই সম া সমাধােনর িট উপায় দখা যাক।

থম উপায় হেলা এটা িব াস করা য উ র একিট polynomial হেব। কত degree এর তা জািন না। তেব polynomial হেবই। এজ আমরা আসেল কেয়কিট n এর মােনর জ উ র বর করব। ভাল হয় যিদ তুিম *n* = 1*,*2*,...*6 এরকম কত িল n এর জ যিদ উ র বর করেত পােরা। আিম জািন n এর এরকম বড় মােনর জ উ র হােত হােত বর করা ক কর িক করার িকছুই নই। তুিম যিদ একটু ক কের এই মান িল বর করেত পােরা তাহেল সমাধােনর অেনক কােছ চেল আসেব। n এর যত িল মােনর জ উ র বর কেরছ িঠক তত িড ী এর polynomial নাও, এে ে

যমন 6 িড ী এর *p*(*n*) = ∑*ai × ni*. যেহতু আমরা *n* = 1*...*6 এর জ *f*(*n*) জািন সেহতু আমরা যিদ *n* এর মান আর *f*(*n*) এর মান বসাই তাহেল 6 িট variable (*a*0*,...a*5) এর 6 িট equation পেয় যাব। এবার gaussian elimination ব বহার কের আমরা *a*0 *...a*5 সবার মান বর কের ফলেত পারব। তামরা চাইেল Lagrange এর interpolation ফমু লাও ব বহার করেত পােরা। ফমু লািট আিম আর এখােন তুেল িদলাম না। কারণ আিম মেন কির এিট একটু advanced িজিনস। যিদ কােরা আ হ থােক তাহেল wiki দখেত পােরা। হয়েতা পের কােনা এক সময় এটা িনেয় িলখব।

ি তীয় উপায় হেলা euler's formula- *V −E* + *F* = 2 যখােন V হেলা vertex এর সংখ া, E হেলা edge এর সংখ া আর F হেলা face এর সংখ া। Face মােন আমােদর সম ায় যই "ভাগ" বলা হেয়েছ সটা। তেব খয়াল রাখেত হেব face ধু ভতেরর অংশ না বািহেরর অংশও িনেত পাের। একটা উদাহরণ দয়া যাক তাহেলই পির ার হেব। একিট ি ভুজ নাও। ি ভুেজর িতনিট vertex, িতনিট edge এবং ইিট face একিট বািহের আর আেরকিট ভতের। তাহেল ফমু লা খােট িকনা

৫৫

দেখা 3*−* 3+2=2. র মত খােট। এবার একিট বগ আঁক আর তার িট কণ আঁক। আমােদর vertex িক 5 টা। বেগ র চার কাণা আর কেণ র ছদ িব । Edge িক 8 িট। বেগ র চার বা আর কেণ র ছদ িব হেত বেগ র কাণা িলেত। আর face িক 5 িট। বেগ র ভতের 4 ভাগ আর বািহের এক ভাগ। তাহেল 5 *−* 8+5=2. বুঝেতই পারছ এই ম ািজকাল ফমু লা আমােদর জীবনেক কত সহজ কের িদেত যাে ! আমােদর সম ায় আমরা যিদ vertex আর edge এর সংখ া বর করেত পাির, তাহেল এই ফমু লা ব বহার কের আমরা face এর সংখ া খুব সহেজই বর কের ফলেত পারব (ভুেল যও না এই ফমু লা িক বািহেরর face কও count কের)। থেম তুিম িচ া কর কীভােব িব িল বসাবা। আইিডয়া হেলা এমন ভােব বসাবা যন এমন কােনা িব যন না থােক যখােন িদেয় 2 িটর বিশ ellipse এর কণ যায় (ellipse এর কণ বলেত বুঝাি আমােদর বসােনা িব

দর মােঝ আঁকা রখা িল)। কারণ যিদ কােনা িব িদেয় িতনিট কণ যায় আমরা চাইেল একিট িব েক সিরেয় একিট ভাগ বাড়ােত পাির। তুিম একটু কাগেজ কলেম এেক দেখা। মেন কর একটা X এর মােঝ িদেয় I এঁেকছ। আর আেরক জায়গায় িঠক মাঝ িদেয় না এেক একটু সিরেয় এঁেকছ। এই ইভােব আঁকা ছিবর face এর সংখ া েন দেখা তাহেলই বুঝেব। অব আমােদর বসােনা n িট িব েত যত খুিশ কণ িমিলত হাক আমােদর সম া নই। তাহেল হল vertex কয়িট? একিট ছিব আঁকা যাক।

1 2

3

4

5 6

9

নকশা ৪.৪: Ellipse এর উপর *n* = 9 িট িব

িচ ৪.৪ এ আমরা ellipse এর উপর 9 িট িব এঁেকিছ। মেন কর আমরা জানেত চাই 1 *−* 5 এই লাইেনর উপর 1 আর 5 ছাড়া আর কয়িট িব আেছ? সহজ, খয়াল কর এই লাইেনর বােম আেছ 3 িট িব আর ডােন আেছ 4 িট িব । তাহেল এেক মাট 3 *×* 4 িট জায়গায় ছদ করেব। এর মােন কােনা একিট রখার বােম যিদ *i* িট িব থােক তাহেল ডােন থাকেব *n − i −* 2 িট িব । তরাং এিট মাট *i*(*n − i −* 2) িট ছদ িব তির করেব। িক 1 থেক লাইন যিদ আমরা 2 এ টানতাম তাহেল *i* = 0. যিদ 3 এ টানতাম তাহেল *i* = 1 এভােব তরাং একিট িব হেত যত িল কণ তির হয় তােত মাট ছদ িব যিদ 9 এ র সংখ া টানতাম ∑*n−*2

*i*=0 তাহেল *i* = 7. *i*(*n − i* + 2). যেহতু একিট িব থেক উৎপ কণ িলেত এেতা িল ছদ িব পাওয়া যায় সেহতু n িট িব হেত উৎপ কণ িলেত ছদ িব র সংখ া এর n ন হেব। তেব একটা িজিনস, এখােন িক over count হেয়েছ। কত বার over count হেয়েছ? িতিট ছদ িব িক িট কেণ র উপর থােক। িতিট কেণ র া িব র জ এই ছদ িব count হেব। অথ াৎ যিদও AB আর CD িট কেণ র একিট ছদ িব , িক সিট A, B, C ও D এই চারিট িব সােপে ই count হেব। তরাং িতিট ছদ িব 4 বার count হেয়েছ। তাই আমােদরেক sum িটেক 4 িদেয় ভাগ করেত হেব। সই সােথ খয়াল রেখা আমরা িক া িব িন নাই এখেনা। তরাং ঐ sum ক 4 িদেয় ভাগ কের n এর সােথ যাগ করেল আমরা vertex এর সংখ া পেয় যাব। যেহতু n অেনক বড় আমরা লুপ চািলেয় এই sum বর করেত পারব না। আমােদরেক sum িটর closed form বর করেত হেব। আমরা ৪.১ এ এটার

৫৬

8

7

closed form বর কের দিখেয়িছ।

*V ′* =

*n−*2∑*i*=0

*i*(*n − i* + 2)

=

*n−*2∑*i*=0

*ni − i*2 + 2*i*

=

*n−*2∑*i*=0

*ni −*

*n−*2∑*i*2 +

*n−*2∑2*i*

(৪.১)

*i*=0

*i*=0

= *n*

এবার E বর করা যাক। আমরা E একটু সহেজ বর করব। আমরা যিদ আমােদর ােফর সব vertex এর degree এর যাগফল বর করেত পাির তােক 2 িদেয় ভাগ করেলই E পেয় যাব। এই সম ায় degree এর যাগফল বর করা সহজ। কারণ িতিট ছদ িব র degree 4 আর শীষ িব

িলর degree *n*+1 (িচে 1 এর সােথ 2 হেত কের 9 সবার একিট edge আেছ। সই সােথ 1 এর সােথ িট ellipse এর border আেছ, এ িলও edge, অথ াৎ ellipse এর border বরাবর 1 য 2 আর 9 এর সােথ যু তার কথা বলিছ)। তরাং এেদর সবাইেক যাগ কের 2 িদেয় ভাগ করেলই আমরা E পেয় যাব। এখন আমরা euler এর ফমু লায় V ও E এর মান বিসেয় F এর মান

বর করেলই আমােদর সম া সমাধান হেয় যােব।

একটা িজিনস। n এর মান িক 231 এর মত হেত পাের। আবার আমােদর ফমু লােত *n*3 এর টাম আেছ। আমরা যিদ long long এও িহসাব কির তাহেলও overflow হেব। তরাং আমরা চাইেল c++ এ bigint এ কাড করেত পাির অথবা java ত কাড করেল আমরা java এর built in library এর BigInteger ব বহার করেত পাির। আিম বলব তামােদর উিচত c++ ও java ই ভােবই সমাধান করা। ই উপােয়র সােথ পিরিচত থাকা ভাল।

UVa 1363 Joseph's Problem

**সমস া:** n ও k দওয়া থাকেব (1 *≤ n, k ≤* 109). ∑*ni*=1 *k* mod *i* বর করেত হেব।

**সমাধান:** সাধারণত তুিম যিদ কাথাও i এর উপর িবশাল লুপ দেখা আর অব ায় থােক (integer division) তাহেল বিশর ভাগ সময় একিট র লুেপর *√n* ভতের i যিদ ভাগ মথড কােজ লােগ। আমরা i এর মান 1 হেত 30 এর জ 30/*i* এর মান টিবল ৪.১ এ ল কির।

যখন তুিম ছাট ছাট সংখ া িদেয় ভাগ কেরছ তখন িনত নতুন মান পাওয়া যায়। যমন 1 িদেয় ভাগ করেল 30, 2 িদেয় করেল 15 আবার 3 িদেয় করেল 10. িক যিদ বড় মান িদেয় কর তখন দখা যায় অেনক িলর জ একই মান আেস। যমন 16 হেত 30 যাই িদেয় 30 ক ভাগ কর না কন 1 হেব। 11 হেত 15 যাই িদেয় ভাগ কর না কন 2 হেব। তরাং মূল আইিডয়া হেলা তুিম i এর মান ক ই ভােগ ভাগ করেব ছাট আর বড়। ছাট মান িল িদেয় ভাগ কের কের *k*/*i* এর মান উ বর েরর করেব। উপর আর লুপ বড় চালােব। িল িদেয় এখন ভাগ ছাট করেব বড় ভাগ না। বরং করেব বড় কীভােব? িল িদেয় সাধারণত ভাগ করেল *√k* যই িদেয় উ র আেস সই

এর সােপে আমরা যত *k*/*i >* ণ *√k*/*i* ছাট *k*. আর এর মান বড় ভাগ *√k* আলাদা এর কের থািক। ছাট i বড় হয়। তাহেল আমরা *n−*2∑*i − n−*2∑*i*2 + 2

*n−*2∑*i*

*i*=0

*i*=0

*i*=0 = *n*(*n −* 2)(*n* 2 িলর জ আমরা 1 হেত তত ণ লুপ চালাব সই সব i েসস কের ফেলিছ যােদর জ

৫৭ *−* 1)

*−* (*n −* 2)(*n −* 6 1)(2*n −* 3)

+ 2(*n −* 2)(*n* 2

*−* 1)

i 30/i i 30/i i 30/i 1 30 11 2 21 1 2 15 12 2 22 1 3 10 13 2 23 1 4 7 14 2 24 1 5 6 15 2 25 1 6 5 16 1 26 1 7 4 17 1 27 1 8 3 18 1 28 1 9 3 19 1 29 1 10 3 20 1 30 1

সারনী ৪.১: 30/i এর টিবল

এবার বড় i িল েসস করা যাক। আমােদর ল হল সসব i েসস করা যােদর জ *k*/*i ≤ √k* হয়। মেন কর আমােদর উ র হল j (অথ াৎ আমরা j এর একিট লুপ চালাি 1 হেত *√k* পয )। তুিম িক বলেত পারেব কান কান মােনর জ j উ র হেব? অথ াৎ আমরা যিদ decimal division এর কথা ভুেল যাই তাহেল আমরা সসব a চাই যন *k*/*a* = *j* হয়। অথ াৎ *a* = *k*/*j*. িকছু উদাহরণ দখা যাক। *j* = 1 হেল 30/1 = 30 যিট হেলা 1 উ র হবার শষ মাথা (কারণ 31 িদেয় ভাগ করেল 0 হেয় যায়)। আমরা যিদ *j* = 2 িদেয় ভাগ কির তাহেল পাই 30/2 = 15 (16 িদেয় ভাগ করেল 1 হেয় যায়) এবং এিট হেলা 2 উ র হবার শষ মাথা। *j* = 3 িদেয় ভাগ করেল 30/3 = 10 পাই এবং 10 হল 3 উ েরর শষ মাথা। তরাং আমরা উ র এর উপর যিদ লুপ চালাই এবং তা িদেয় যিদ k ক integer ভাগ কির তাহেল আমরা সই উ র এর শষ মাথা পেয় যাব। তেব এই কাজ *√k* এর থেক বিশ উ েরর জ করেত যও না (যিদ j এর লুপ িঠক মত চািলেয় থাক তাহেল এই সম া হেব না)। বা করেলও একটু সাবধােন কিরও। যমন আমােদর উদাহরেণ যিদ আমরা উ র 9 ধরেত চাই এবং 30/9 = 3 কির তাহেল িক হেব না। 3 এ িক উ র 10. আসেল 9 উ র আলা কােনা সংখ াই নই। তুিম যিদ একটু িচ া ভাবনা কর তাহেল বুঝবা য *√k* পয িতিট মান উ র হবার যাগ । তাই এ িনেয় আমরা ওত িচ া করব না। যাই হাক আমরা যিদ জািন 1 উ র হেব শষ কান সংখ ায় (30), 2 উ র হেব শষ কান সংখ ায় (15), 3 উ র হেব শষ কান সংখ ায় (10) তাহেল িক আমরা কান উ র কয়বার বা কান কান সংখ ার ে কানিট উ র আসেব সব বর কের ফলেত পারব। যমন আমরা বলেত পাির 1 উ র হেব 16 হেত 30, 2 উ র হেব 11 হেত 15 ইত ািদ। আমরা চাইেল এখন যই সব সংখ ার জ j উ র তােদর count িনেত পাির, তােদর

যাগফল িনেত পাির ইত ািদ।

এত ণ আমরা একটা সমাধােনর টকিনক বললাম।এই সমাধােনর টকিনক ব বহার কের আমরা আমােদর সম া কীভােব সমাধান করা যায় তা দখেবা। থমত খয়াল কর, আমােদর সম ায় লুেপর ভতের mod আেছ। এেক আমরা ভােগ পিরবত ন করেত পাির। *k* mod *i* = *k −* (*k*/*i*)*i* যখােন ভাগিট হেলা integer division. তরাং আমরা িলখেত পাির ∑*ni*=1 *k −* (*k*/*i*)*i* বা আসেল আমােদর ∑*ni*=1 (*k*/*i*)*i* বর করেলই চেল কারণ ∑*ni*=1 *k* = *nk*। এবার আমােদর টকিনক কােজ লাগেব। থেমই দেখ নাও n িক k এর থেক বড় িকনা। বড় হেল *i* = *k*+1 হেত *n* এর জ *k*/*i* য 0 হেব তােতা আমরা আেগ থেকই জািন। তরাং সসব বাদ। আমরা ধের িনেত পাির *n ≤ k*. এখন আমরা ছাট বড় ই ভাগ করব *√k* এর িভি েত। থেম *i* = 1 হেত লুপ চািলেয় (*k*/*i*)*i* বর করেত থাকব যত ণ *k*/*i > √k* হয় ( খয়াল করেত হেব i যন আবার n এর থেক বড় না হেয় যায়)। এবার ভাগফেলর উপর লুপ চালােনার পালা। আমরা এবার j এর লুপ চালােবা 1 হেত *√k* পয । িতবার j উ র হেব এরকম শষ সংখ া বর করব *k*/*j*. *j* + 1 উ র হেব এরকম শষ সংখ া *k*/(*j* +1). তরাং *k*/(*j* +1)+1 হেত *k*/*j* পয সংখ া িলর জ *k*/*x* = *j* হেব যখােন x হেলা ঐ রে থাকা যেকােনা সংখ া। তরাং আমােদর sum য িছল (*k*/*i*)*i* এর উপর সটা এখন পালেট হেয় যােব *jx* কারণ আমরা ধেরিছ j হল ভাগফল আর x হেলা সরকম সংখ া িল।

৫৮

তরাং িত j এর জ আমরা যিদ x িল অথ াৎ *k*/(*j* +1)+1 হেত *k*/*j* পয এর যাগফল বর কের ফলেত পাির তাহেলই হেব। আর এই যাগফল চাইেলই ফমু লার সাহােয বর কের ফলা যায়। একটা িজিনস আবােরা খয়াল রাখেত হেব যন এই য সংখ ার র এর মােঝ যন n এর থেক বড় সংখ া না থােক। নেত কিঠন লাগেলও আসেল এটা ওত কিঠন না। তুিম একটু কাগেজ কলেম িনেজ

থেক িচ া কের দেখা আর কাড কের দেখা তাহেলই পির ার হেয় যােব।

UVa 11440 Help Mr. Tomisu

**সমস া:** M ও N িট সংখ া দওয়া হেব (1 *≤ M ≤ N ≤* 107 এবং *N − M ≤* 105). 2 এবং N! (N factorial) এর মােঝ কত িল সংখ া আেছ যােদর সকল prime divisor ই M হেত বড়? 500 িট test case থাকেত পাের।

**সমাধান:** এধরেনর mathematical সম ার ে সমাধােনর কােনা িনিদ টকিনক নই। মূলত observation এর দরকার। অথ াৎ কাগজ কলম িনেয় এিদক অিদক নানা িদক থেক িচ া কের সমাধান করেত হেব। এই সম ার ে মূল observation হেলা 2 থেক N! এর মােঝ যসকল সংখ া M! এর সােথ coprime তারাই উ র। কন? কারণ M! এর ভতের M এর থেক ছাট সব prime আেছ। আবার M এর থেক বড় কােনা prime এেত নই। তরাং যসকল সংখ া M! এর সােথ coprime তারাই উ র। *phi*(*M*!) িক বর করা সহজ তাই না? এিট হেলা *M*!(1 *−* 1/*p*1)(1 *−* 1/*p*2)*...* যখােন *pi* হেলা M এর সমান বা ছাট prime. তরাং আমরা যিদ 107 পয সকল prime generate কের ফিল তাহেল খুব সহেজই 107 পয সকল সংখ ার factorial এর phi function আমরা বর কের ফলেত পারব। কীভােব?

খয়াল কর *phi*((*i −* 1)!) এর সােথ *phi*(*i*!) এর খুব close স ক আেছ। থমত ফমু লার থম term িট (*i −* 1)! হেত *i*! হেয় যােব। সই সােথ যিদ *i* prime হয় তাহেল (1 *−* 1/*i*) নােমর একটা term চেল আসেব। অথ াৎ যিদ i prime না হয় তাহেল *phi*((*i −* 1)!) এর সােথ i ন করেলই *phi*(*i*!) পাওয়া যােব। আর যিদ prime হয় তাহেল (*i −* 1) ন করেলই হেব। এভােব আমরা *phi*(*i*!) precalculate কের ফলেত পারব। িক এটােতা বর হেলা *M*! এর সমান বা ছাট কয়িট coprime সংখ া আেছ (M! এর সােথ)। আমােদর তা দরকার *N*! এর সমান বা ছাট কয়িট coprime আেছ (M! এর সােথ)। আসেল *phi*(*M*!) ক তামরা যিদ *N*!/*M*! িদেয় ন কর তাহেলই হেব। কন? দেখা 1 হেত M! এর মােঝ x যিদ M! এর সােথ coprime হয় তাহেল (2*M*! + *x*) ও coprime হেব, (3*M*! + *x*) ও হেব। অথ াৎ 1 হেত M! এর মােঝর coprime এর িল M!+1 হেত 2M!, 2M! + 1 হেত 3M! ইত ািদর মােঝও থাকেব। তরাং আমােদর *phi*(*M*!) এর সােথ *N*!/*M*! বা (*M* + 1)(*M* + 2)*...N* ন করেত হেব। যেহতু *N −M ≤* 105 তাই আমরা িত case এ এই কাজ করেতই পাির।

UVa 10214 Trees in a Wood

**সমস া:** তামােক [*−a, a*] *×* [*−b, b*] সাইেজর একিট ি ড দওয়া থাকেব যখােন 1 *≤ a ≤* 2000*,*1 *≤ b ≤* 2 *×* 106. তুিম যিদ (0*,*0) ত দাঁড়াও তাহেল তামােক বলেত হেব কয়িট ি ড িব দখেত পারেব। যমন, তুিম যিদ (0*,*0) ত দাঁড়াও তাহেল তুিম (1*,*1)*,*(*−*1*,*1) এেদর দখেত পারেব িক (2*,*2)*,*(10*,*10)*,*(5*,*0) এেদর দখেত পারেব না। সম ার িবধার জে ধের নাও (0*,*0) ক আমরা গণার ভতর ধরব না অথ াৎ এই িব িট দখা গল িক গল না তা িনেয় না ভাবেলও হেব।

**সমাধান:** মূল সম ায় এই count ক মাট ি ড িব র সংখ া িদেয় ভাগ কের output করেত হেব, িক আমার মেন হয় তামরা সবাই মাট ি ড িব র সংখ া িনেজরাই নেত পারেব। তরাং এখােন আমরা কয়িট িব ক দখেত পারব সটার উপর নজর দই। থম কথা, আমরা চার axis এ চারিট িব দখেত পারব (1*,*0)*,*(0*,*1)*,*(*−*1*,*0)*,*(0*,−*1). আমরা যিদ কােনাভােব বর করেত পাির য

৫৯

থম quadrant এ (axis বরাবর িব বােদ) কয়িট ি ড পেয় টেক আমরা দখেত পারব, তাহেল তােক চার িদেয় ন করেলই আমরা সকল quadrant এর জ উ র বর কের ফলেত পারব। কারণ সব quadrant হেলা symmetric. তরাং আমরা জানেত চাই [1*,a*] *×* [1*,b*] এর মােঝ কান কান িব েক আমরা দখেত পারব। ধরা যাক আমরা (*i, j*) ক দখেত পাই। তাহেল িক (*ki, kj*) ক আমরা দখেত পারব না। আবার অ ভােব বলেল বলা যায় যিদ (*i, j*) একিট িব হয় এবং i ও j এর যিদ একিট সাধারণ ণনীয়ক g থােক তাহেল (*i*/*g,j*/*g*) ারা (*i, j*) বাধা া হেব। অথ াৎ আমরা সসব (*i, j*) ক দখেত পারব যােদর কােনা সাধারণ ণনীয়ক নই। তার মােন আমােদর বর করেত হেব 1 *≤ i ≤ a,*1 *≤ j ≤ b* হেল কয়িট (*i, j*) এর জ *gcd*(*i, j*)=1 বা i ও j পর র coprime হেব। আিম এখােন এিট সমাধােনর ইিট উপায় িব ািরত বলব আর আেরকিট advanced উপােয়র জ সামা hints দব। যােদর ই া তারা যন নেট দেখ িনেত পাের।

থম উপায় হেলা euler এর phi function ব বহার করা। িকছু ণ আেগই আমরা দেখিছ i এর সমান বা ছাট এবং তার সােথ coprime এরকম সংখ ার count হেলা *phi*(*i*). আর িকছু ণ আেগই (UVa 11440) এ দেখিছ এই coprime িল িত i িট সংখ ায় repeat হয়। অথ াৎ 1 হেত i এর মােঝ যত িল সংখ া i এর সােথ coprime, *i*+1 হেত 2*i* এর মােঝ িঠক ততিট সংখ াই coprime, 2*i*+1 হেত 3*i* এর মােঝও। এভােব চলেত থাকেব। তরাং আমরা i এর একিট লুপ চালােবা 1 হেত a পয । এবং িতিট i এর জ আমরা দখেবা 1 হেত b এর মােঝ কত িল সংখ া আেছ যারা i এর সােথ coprime. এজ আমরা থেম *phi*(*i*) বর করব এবং *b*/*i* িদেয় (integer division) ন করব। বািক থাকেব *i*(*b*/*i*)+1 হেত b পয সংখ া িল। এখােন বািক সংখ ার পিরমাণ *b* mod *i* তাই না? কারণ এ িল হেলা i এর সবেথেক শষ multiple থেক b পয সংখ া িল। যেহতু i এর মান a থেক বড় না তরাং এর মান 2000 থেকও বড় না। অথ াৎ এই য অিতির সংখ া তা কখনই 2000 থেক বড় হেব না। আর এিট িতিট i এর জ ই। তরাং আমরা বলেত পাির 20002 িট সংখ ার জ আমােদরেক আলাদা কের gcd বর করেত হেব। এটা িত case এ কােনা ব াপার না। যিদও কয়িট case থাকেব সটা আমােদর বলা নই। তাহেল আমােদর এই সমাধােনর time complexity হেব O(*a*2). সমাধানটা হেলা- আমরা i এর একিট লুপ চালােবা 1 হেত a পয । তার phi(i) বর করব এবং *b*/*i* িদেয় ন করব। এরপর *i*(*b*/*i*)+1 থেক b পয লুপ চািলেয় i এর সােথ gcd বর কের দখেবা য তােদর gcd 1 িকনা। এভােব gcd 1 ওয়ালা pair এর সংখ া count করব।এবার ি তীয় উপায় বলা যাক। যখনই দখবা coprime হেত হেব বা তােদর gcd g হেত হেব এরকম ধরেনর কথাবাত া বলা থাকেব তখন এই প িত ায়ই কােজ লােগ। যিদ বলত *gcd*(*i, j*) = *g* কত িল (*i, j*) এর জ হেব? তাহেল িজিনসটা কিঠন। িক যিদ বলত *gcd*(*i, j*) g এর multiple হেব কত ভােব? তাহেল িক সই উ র বর করা সহজ। এিট িক (*a*/*g*)*×*(*b*/*g*) (integer ভাগ)। কন? কারণ [1*,a*/*g*] *×* [1*,b*/*g*] এই ি েডর যেকােনা িব ধরা যাক (*x, y*) তাহেল (*gx,gy*) িব িট [1*,a*] *×* [1*,b*] ি েড থাকেব এবং তােদর gcd হেব g এর multiple. এখন একটা িজিনস িচ া কর [1*,a*] *×* [1*,b*] এই ি েড যেকােনা (*x, y*) এর gcd িক সেব া *M* = *min*(*a, b*)। তাহেল আমরা 1 হেত M পয িতিট সংখ া i এর জ বর করেত পাির কত িল িব আেছ যােদর gcd i এর multiple. তাহেল আমােদর সম া িনেচর সম ায় পিরণত হেব-

f ও g িট n সাইেজর অ াের দওয়া আেছ। *f*[*i*] এ থাকেব g এর অ ােরেত i এর multiple এর পিজশন িলেত যসব সংখ া থাকেব তােদর যাগফল অথ াৎ *f*[*i*] = ∑*j g*[*j*] যখােন j হেলা i এর multiple. f এর মান সমূহ দওয়া আেছ, g এর মান িল বর করেত হেব।

এিট একিট common সম া। অথ াৎ িবিভ সম া সমাধান করেত গেল এরকম সম ায় reduce হেয় চেল আেস। এই সম া *n*log *n* এ সমাধান করা যায়। আইিডয়া হল g এর মান পছন থেক বর করা, অথ াৎ থেম *g*[*n*], এরপর *g*[*n −* 1], এরপর *g*[*n −* 2] এরকম কের এেক এেক g[1] পয মান বর করেত হেব। মেন কর *g*[*i*] এর মান বর করেত চাও। যেহতু আমরা পছন থেক মান বর করিছ তরাং i এর থেক বড় সব index এ g এর মান ইেতামেধ ই বর কের ফেলিছ। অথ াৎ আমরা i বােদ i এর সব multiple এর মান বর কের ফেলিছ। এখন আমরা যিদ *f*[*i*] হেত i বােদ i এর অ া multiple এর index এ থাকা g এর মান িবেয়াগ কির তাহেল *g*[*i*] বর হেয় যােব। অথ াৎ তুিম যখন *g*[4] বর করেত চাইেব তখন যা করেত হেব তাহেলা *g*[4] = *f*[4]*−g*[8]*−g*[12]*−g*[16]*...*. আর এটার complexity য sieve এর মত হেব তােতা

৬০

বুঝেতই পারছ।

তৃতীয় উপােয়র কথা আিম িব ািরত বলব না। যােদর আ হ আেছ তারা mobious inversion পেড় দখেত পােরা। এভােব করেল complexity িত কেস O(*n*) হেব (যিদও sieve এর মত precalculation লাগেব একটা)।

UVa 1393 Highways

**সমস া:** একিট *n × m* সাইেজর ি ড দওয়া আেছ (1 *≤ n, m ≤* 300)। মেন কর ি েডর িতিট িব এেককিট শহর। এখন তুিম িতিট শহেরর pair এর মােঝ রা া বানােব। এর ফেল িকছু রা া overlap করেত পাের, যমন (1*,*1) হেত (2*,*2) পয রা া (1*,*1) হেত (3*,*3) পয রা ার সােথ overlap কের। এসব ে ছাট রা ােক িবেবচনা করার দরকার নই। তামােক বলেত হেব মাট কত িল রা া আেছ।এই গণনার সময় horizontal বা vertical রা া িলও িবেবচনার বািহের রেখা।

**সমাধান:** আমরা ইেতামেধ ই িশেখ এেসিছ য যিদ i ও j এর gcd 1 হয় তাহেল (0*,*0) হেত (*i, j*) পয রখা টানেল সিট কােনা িব র উপর িদেয় যায় না। অথ াৎ এই রা া অ কােনা ছাট রা ােক overlap কের না। আমরা এরকম সব ছাট ছাট রা া নব তাহেলই আমােদর মূল সম া সমাধান হেয় যােব। আসেলই িক তাই? না। খয়াল কর (0*,*0) *−* (1*,*1) এবং (1*,*1) *−* (2*,*2) এ রা া িট কাউেকই cover কের না। িক এ িটেক আলাদা কের count করেল overcount হেয় যােব। তাহেল িক করা যায়? আমরা িক সবেচেয় বড় রা া িল count করব? আমরা যিদ তা করেত পারতাম তাহেল আসেলই এই সম া সমাধান হেয় যত। িক আিম এভােব ভেব কােনা সমাধান আনেত পাির নাই।

যভােব সমাধান হেব তাহেলা িত direction এর জ বর করা য কত িল রা া আেছ যা এই বরাবর চেল। যমন একিট 3 *×* 3 ি েড কত িল (1*,*1) direction এর রা া আেছ? 3 িট। (0*,*0)*,*(1*,*0)*,*(0*,*1) এই িতনিট িব হেত। অ িব িল হেত তুিম যিদ (1*,*1) বরাবর রা া টান তাহেল হয় তা সরাসির বাইের যায় ( যমন (2*,*0) হেত) অথবা আেগর িতনিট রা ার উপর থােক ( যমন (1*,*1) হেত)। তাহেল আমরা i ও j এর উপর লুপ চালােবা। যিদ দিখ তােদর gcd 1 না তাহেল তা বাদ। আর নাহেল আমােদর বর করেত হেব এই িদক বরাবর কত িল িভ িভ রা া আেছ। বা অ ভােব কত িল িব থেক এই রা া হেত পাের? ধরা যাক আমরা জানেত চাইিছ (*x, y*) হেত এই রা া হেত পাের িকনা। এখান থেক যিদ কােনা রা া হয় তাহেল এর পেরর িব হেব (*x*+*i, y*+*j*). আর আেগই বেলিছ এই রা া অ কােনা রা ােক overlap করেত পারেব না। তাই আমরা (*x−i, y−j*) ও দখেবা। কারণ যিদ (*x−i, y−j*) িব িট আমােদর ি েড থােক তাহেল ঐ িব থেক (*i, j*) িদেকর রা া (*x, y*) হেত একই িদেকর রা ােক overlap করেব। তাহেল (*x, y*) হেত রা া হেত পারেব যিদ (*x* + *i ≤ m* এবং *y* + *j ≤ n*) এবং (*x − i <* 0 বা *y − j <* 0) হয়। এরকম কত িল (*x, y*) পাওয়া যােব? এরকম ে যটা করা উিচত তাহেলা একটা ছিব আঁকা এবং valid অংশ িচি ত কের সই অংেশ কত িল িব আেছ তা count করা। এই অংশ তামােদর উপর ছেড় দয়া যায়।

তাহেলএভােব আমরা িত case এ O(*nm*) এ সমাধান করেত পাির। মেন হয় এই complexity ত accepted হেয় যােব। তেব তামরা যারা সমাধানেক আরও ভাল করেত চাও তারা িচ া কের O(3002)এ precalculation কের রাখাযায় িকনা চ া কের দখেত পােরা।সমাধানটােত dynamic programming ব বহার করেত হেব। এই সমাধানটা যত না math স িক ত তার থেক অেনক বিশ dynamic programming স িক ত। তাই এখােন আর এ িনেয় আেলাচনা বাড়ালাম না।

UVa 1642 Magical GCD

**সমস া:** n িট ধনা ক সংখ া দওয়া আেছ যখােন n সেব া 105 এবং িতিট সংখ া সেব া 1012 হেত পাের। এমন একিট contiguous subsequence বর করেত হেব যার element িলর gcd

৬১

এবং সই subsequence এর length এর ণফল যন সকল contiguous subsequence এর মােঝ সবেচেয় বিশ হয়। যমন- যিদ আমােদর মূল sequence হয় 30*,*60*,*20*,*20*,*20 তাহেল আমােদর উ র হেব 80 যা আমরা শষ চারিট element িদেয় তির contiguous subsequence

থেক পােবা।

**সমাধান:** এখােন সবেচেয় বড় observation হেলা-

মেন কর তুিম i এ শষ হয় এরকম সকল contiguous subsequence িনেল এবং তােদর gcd িল বর করেল। এখােন আসেল log*a* এর বিশ িভ িভ gcd নই যখােন a হল আমােদর i তম সংখ া। অথ াৎ তুিম যিদ (*i*)*,*(*i −* 1*,i*)*,*(*i −* 2*,i −* 1*,i*)*,...*(1*...i*) এরকম index িলর জ যিদ যথা েম *gcd*(*a*[*i*])*, gcd*(*a*[*i−*1]*,a*[*i*])*...gcd*(*a*[1]*,...a*[*i*]) বর কর তাহেল এখােন log*a*[*i*] এর থেক বিশ িভ gcd থাকেব না।

কন? কারণ খুব সহজ। মেন কর তুিম i এ শষ হওয়া সকল contiguous subsequence এর জ gcd বর করেল। অথ াৎ *gcd*(*a*[*i*]), *gcd*(*a*[*i*]*,a*[*i −* 1]) এভােব *gcd*(*a*[*i*]*...a*[1]) পয । এরা একিট non increasing sequence. কারণ আেগর সংখ ার সােথ নতুন আেরকিট সংখ া লািগেয় gcd বর করেল নতুন সংখ া কখনই আেগর সংখ া থেক বড় হেব না। আর যিদ ছাট হয় তাহেল কমপে ই ন ছাট হেব। কারণ পেরর সংখ া তা আেগর সংখ ােক ভাগ দেবই। আর যিদ সই সােথ ছাট হেত হয় তাহেল কমপে ভােগর সমান হেব। তার থেক ছাটও হেত পাের, িক বড় হবার স াবনা নই। অথ াৎ যখন ছাট হে তখন অ ত 2 ভাগ হেবই। তরাং আমােদর এখােন আসেল log(*a*[*i*]) এর থেক বিশ িভ gcd থাকেব না। কারণ তুিম যিদ *a*[*i*] ক log*a*[*i*] বার 2 িদেয় ভাগ কর তাহেল তা 1 হেয় যােব। আর একবার যিদ 1 হেয় যায় তাহেল তা আর ছাট হেত পারেব না।

তাহেল কীভােব আমরা এই সম া সমাধান করব? খুব সহজ। আমরা একিট অ ােরেত িভ িভ gcd এর জ িলেখ রাখেবা য এই gcd পেত আমরা সেব া এেতা দূর যেত পাির। অথ াৎ g gcd পেত আমরা i হেত j পয যেত পাির (*j<i*), g' পেত হয়েতা k পয যেত পাির (*k<j*). এরকম। যখন আমরা নতুন সংখ া অথ াৎ *i* + 1 তম সংখ া নব তখন আমরা নতুন সংখ ার সােথ পুরাতন gcd (g, g' ইত ািদ) িলর gcd বর করব। যিদ পাশাপািশ িট gcd একই হয় তাহেল ঐ ইিট range ক merge কের দব। এভােব চলেত থাকেব। উদাহরণ দওয়া যাক, মেন কর আমােদর sequence এ িট সংখ া আেছ *a*[0] = 25*,a*[1] = 60. তাহেল আমােদর অ ােরেত িলখা থাকেব 60 পেত 1 পয যেত হয়, আর 5 পেত 0 পয যেত হয়। এখন মেন কর নতুন সংখ া আসল 35 (*a*[2] = 35). তখন আমরা িলখব 35 পেত 2 পয যেত হয়। 5 পেত যেত হয় 1 (gcd(35, 60) = 5), 5 পেত যেত হয় 0 (gcd(35, 5) = 5). খয়াল কর শষ িট gcd একই, তরাং এেদর merge কের আমরা িলখেত পাির 5 পেত 0 পয যেত হয়। এই কাজ আসেল খুব সহেজই কাড করা যােব, স িনেয় িব ািরত বলিছ না। এই সম া সমাধােনর মূল িজিনস হেলা log*n* সং া observation.

UVa 10868 Bungee Jumping

**সমস া:** w ভেরর এক জন লাক একিট ি জ এর উপর হেত bungee jumping করেব। ি জিট s উ তায় আেছ। এখন লাকিট যিদ লাফ মাের তাহেল স একিট free fall body িহসােব কাজ করেব, অথ াৎ স কােনা বাধা ছাড়া পৃিথবীর িদেক আসেব এবং তার accelaration হেব g (অিভকষ জ রণ). দিড়িটর দঘ l. যখন তার দূর ি জ হেত l এর থেক বিশ হেয় যােব তখন bungee এর দিড় তােক উপের টানেব। ধরা যাক স l দূরে র থেকও ∆*l* দূর বিশ অিত ম কেরেছ। সসময় দিড়িট উপেরর িদেক *k*∆*l* force এ টানেব। লাকিট মািটেত িনরাপদ ভােব পৗঁছােব যিদ মািট েশ র সময় তার বগ 10*ms−*1 এর বিশ নাহয়। যিদ তার বিশ হয় তাহেল লাকিট মের যােব। তামােক *w, l, s, k* এর মান দওয়া আেছ। বলেত হেব লাকিট িক িনরাপেদ মািটেত নামেত পারেব নািক মের যােব নািক মািট পয আসেতই পারেব না।

**সমাধান:** িনঃসে েহ আরও একিট বশ কিঠন সম া। সিত বলেত এই সম া সমাধান করা আমার

৬২

জ ও এক রকেমর physics ঝালাই।

এই সম া সমাধােনর জ আমােদর মূল যই theorem জানেত হেব তাহেলা- শি র কােনা ংস নই, এিট কবল এক প হেত অ েপ পা িরত হয়। এখােন মা ষিট একিট ি েজর উপর দািড়েয় আেছ। তরাং তার একিট potential energy (ি িত শি ) আেছ। এখন স যিদ লাফ দয় তাহেল যত ণ না স মািটেত আসেছ তত ণ তার potential energy কমেত থাকেব। িক এই কমা শি কই যােব? এর এক অংশ পিরণত হেব kinetic energy (গিত শি ) ত আর আেরক অংশ পিরণত হেব bungee দিড়র potential energy ত, কারণ দিড়র দঘ *l* এর থেক বিশ হেয় গেল সিট আবার িপেছ (মােন উপের) টানেত করেব। তরাং আমােদর কাজ এই সব energy বর করা।

তামরা চাইেল potential energy এবং kinetic energy এর ফমু লা বা তার derivation নেট দখেত পােরা। মূল কথা হেলা *m* ভেরর কােনা ব যিদ ভূপৃ হেত *h* উ তায় থােক তাহেল তার potential energy হেব *mgh* যখােন g হেলা অিভকষ জ রণ। আর v বেগর কােনা ব র kinetic energy হেলা 12*mv*2. আর সম ােতই বলা আেছ য bungee দিড় যিদ সাম অব া হেত ∆*l* দূর সরােনা হয় তাহেল সখােন দিড়িট *k*∆*l* শি েয়াগ করেব। তাহেল দিড়র সাম অব া থেক *z* দূরে এর potential energy এর পিরমাণ কত হেব? এভােব িচ া কর x এ থাকেত এর উপর কাজ করা বেলর পিরমাণ *kx*. এখন যিদ তুিম তােক খুব ছাট *dx* দূর সরাও তাহেল কাজ হেব *kxdx* (কাজ = বল x সরণ). তাহেল 0 হেত z পয সড়ােত কােজর পিরমাণ ∫ *z*0 *kxdx* বা 12*kz*2.তাহেল আমরা আমােদর েয়াজনীয় সব ফমু লা জািন। এখন আমােদর কাজ িকছু equation দাঁড় করােনা। েতই বেলিছ শি র ংস নই। তরাং একদম েত যখন লাকিট ি েজর উপের িছল তখন তার potential energy এর পিরমাণ িছল *wgs*. যেহতু স তখন ি র অব ােন িছল তাই তার kinetic energy 0. তরাং েত মাট energy *wgs*.

এখন মেন কর স লাফ িদল এবং *x* দূর অিত ম কের ফলল। এই অব ায় তার potential energy হেব *wg*(*s−x*). িক kinetic energy? এটা তার বেগর উপর িনভ র কের, ধরলাম তার বগ v, তাহেল kinetic energy হেব 12*wv*2. আর bungee দিড়র potential energy? যিদ *x ≤ l* হয় তাহেল িক দিড়র potential energy 0. কারণ এখেনা দিড় তার সাম অব ায় যায় নাই, সাম অব া হেত দূের যাওয়া তা দূেরর কথা। তরাং এই ে আমােদর শি র িনত তা সূ বেল *wgs−wg*(*s−x*) = 12*wv*2 কারণ আমােদর potential energy ত যই পিরমাণ পিরবত ন হেয়েছ তা স ূণ kinetic energy ত পা িরত হেয়েছ, দিড়র potential energy ত পিরণত হয় নাই। এখন যিদ *s ≤ l* হয়, অথ াৎ দিড়র দঘ যিদ ি েজর উ তার চেয় বড় হয় তাহেল আমরা এই সূ হেত v এর মান বর করেত পাির (*x* = *s* বিসেয়)। এবং v এর মান 10 এর সােথ তুলনা কের দখেবা য লাকটা মের যায় নািক ভালভােব মািটেত পৗঁেছ।

এখন যিদ *x>l* হয় তাহেল? এবার তাহেল দিড়র potential energy আেছ। সাম অব া থেক এিট *x−l* দূরে আেছ। তরাং এর potential energy হেব 12*k*(*x−l*)2. অথ াৎ আমােদর equation হেব *wgs − wg*(*s − x*) = 12*wv*2 + 12*k*(*x − l*)2. এখন িক করেব? এই সমীকরেণ *x* = *s* বিসেয় v বর করেব? না। তার আেগ দখেত হেব দিড় আেদৗ মািটেত পৗঁছায় িকনা। যিদ দিড় মািটেত না পৗঁছায় তাহেল মািটেত পৗঁছানর আেগই কাথাও থেক স ফরত িগেয়েছ। িঠক সই জায়গােত সই মুহূেত *v* = 0 হেয় গেছ। তাই উপেরর equation এ *v* = 0 বিসেয় x বর কর। যিদ দেখা য *x<s* এর জ সমাধান আেছ তার মােন স মািট েশ র আেগই িফের গেছ। আর নাহেল মািট শ কেরেছ। যিদ মািট শ কের তাহেল *x* = *s* বিসেয় v এর মান বর কের

ফল।

LOJ 1278 Sum of Consecutive Integers

**সমস া:** একিট পূণ সংখ া N দওয়া থাকেব (1 *≤ N ≤* 1014). বলেত হেব N ক কত ভােব একািধক ধনা ক িমক পূণ সংখ ার (consecutive positive integers) যাগফল িহসােব কাশ

৬৩

করা যায়। যমন *N* = 15 ক আমরা িতনভােব কাশ করেত পাির 1+2+3+4+5, 4+5+6 এবং 7+8. Test case সংখ া 200 িট।

**সমাধান:** এধরেনর সম া িল সমাধান করা খুব একটা কিঠন না। যা করেত হয় তাহেলা এেক ফমু লার আকাের িলখেত হেব এরপর সই ফমু লা কতভােব সমাধান করা যায় তা দখেত হেব। যমন এই সম ােত আমরা ধরেত পাির আমােদর িসিরেজর র সংখ া *a* আর মাট *n* িট সংখ া আেছ সই িসিরেজ। তাহেল আমােদর িসিরজ টা হেব *a, a* + 1*,...*(*a* + *n −* 1). এেদর যাগফল হেব *N*. আর শত মেত *n >* 1 এবং *a >* 0 হেত হেব। এই িসিরেজর সংখ া িলেক যাগ করেল কত হেব? *n*(2*a*+*n−*1)

2 . িবিভ ভােব এই যাগফল বর করা যায়। একিট উপায় হেলা ∑*n−*1

*i*=0 *a* + *i* তেব সহজ উপায় হেলা গাউেসর প িত ব বহার করা। এ িনেয় এর আেগর বইেয় আেলাচনা হেয়েছ তেব যিদ কউ না জেন থােকা নেট দেখ িনেত পােরা। যাই হাক, তাহেল এই যাগফেলর মান হেলা *N*. এর মােন আমরা িলখেত পাির *n*(2*a*+*n−*1)

2 = *N* বা *n*(2*a* + *n −* 1) = 2*N*. আমরা *N* জািন িক *a* বা *n* জািন না। আমােদর ল হেলা কত িল valid *a* ও *n* এর মান পাওয়া যােব তা বর করা। এই ফমু লা থেক একিট িজিনস পির ার য *n* হেব 2*N* এর একিট ণনীয়ক। আর িক িক বুঝা যায়? যেহতু আমােদর ডান পাশ একিট জাড় সংখ া তরাং আমােদর বাম িদেকর িট factor এর একিটেক অব ই জাড় হেত হেব। িক একটু খয়াল করেল তুিম আরও িকছু আিব ার করেব। যমন মেন কর *n* জাড়, তাহেল 2*a*+*n−*1 অবশই িবেজাড় হেব। আবার যিদ *n* িবেজাড় হয় তাহেল 2*a*+*n−*1 অব ই জাড় হেব। উদাহরণ দখা যাক, ধর *N* = 6. তাহেল *n*(2*a*+*n−*1) = 12. আমরা 12 ক নানা ভােব factor করেত পাির 1*×* 12*,*2*×* 6*,*3 *×* 4. এর মােন আমােদর n এই 6 িট সংখ ার কােনা একিট হেবই। িক কান কানিট হেত পাের? তুিম যিদ n এর মান ধর 2 তাহেল 2*a* + *n −* 1 ক হেত হেব 6 (কারণ *n*(2*a* + *n −* 1) = 12) আর তাহেল *a* = 5/2. এরকম কের সকল n এর জ আমরা a এর মান বর করেত পাির। এবং এই মান িলর িদেক তাকােল আমরা বুঝেত পারব কান কান মান n এর জ valid আর কান কানিট না। valid িক না কীভােব বুঝেব? যিদ n ও a িটই পূণ সংখ া হয় এবং *n >* 1*,a ≥* 1 হয় তাহেলই সিট valid. আরও একিট উদাহরণ দখা যাক। ধর *N* = 15. তাহেল *n*(2*a* + *n −* 1) = 30. 30 ক আমরা কত ভােব factorize করেত পাির? 1 *×* 30*,*2 *×* 15*,*3 *×* 10*,*5 *×* 6. তরাং *n* = 1*,*2*,*3*,*5*,*6*,*10*,*15*,*30 এই 8 রকম মান হেত পাের। এর মােন িক N = 15 ত উ র 8? না। আমরা Problem Description এ দেখিছ উ র 3. তাহেল কািহনী িক? থমত *n >* 1 হেত হেব। তাই 1 বাদ। এখেনা 7 িট মান বািক আেছ। তুিম যিদ *n* = 10 চ া কর তাহেল দখেব 2*a*+ *n−* 1=3 আর তাহেল *a* = *−*3. িক a ক ধনা ক হেত হেব। এর মােন *n* = 10 হেত পারেব না। একই ভােব *n* = 6*,*15*,*30 হেত পারেব না। এর মােন সমাধান 3 িট। তাহেল এই ইিট উদাহরণ থেক িক িশখেল? িশখলাম য, 2*N* ক factorize করেলই হেব না। 2*N* এর যত factor আেছ তােদর মােঝ িকছু িকছু *n* হেত পারেব। িক কারা? থমত 1 হেত পারেব না। কারণ *n >* 1. আর *a >* 0 হবার জ আমােদর িক শত পূরণ করেত হেব? একটু িচ া করেল বুঝেব য 2*a*+*n−*1 *> n* হেত হেব। কন? কারণ 2*N* এর একিট factor হেলা *n* আর ওপর factor 2*a* + *n −* 1. যেহতু a ধনা ক তাহেল তা 2*a* + *n −* 1 *> n* হেবই তাই না (কারণ *a >* 0 এবং integer হেল 2*a−*1 *>* 0)? তরাং আমরা সব শেষ যা বুঝেত পারলাম তাহেলা 2*N* এর কত িল divisor *x* আেছ যা 1 এর থেক বড়, *x* ও 2*N*/*x* একই সােথ জাড় না এবং *x <* 2*N*/*x*১ হেত হেব। আমরা যা করেত পাির তাহেলা 2*N* এর িতিট divisor িনেয় িনেয় চক করেত পাির। িক 200 িট case এর জ এই কাজ করা একটু বিশ costly হেয়

যেত পাের। এর থেক িক আরও ভাল কােনা উপায় আেছ?

এই পয ােয় এেস তুিম নানা ভােব িচ া করেত পােরা। একিট উপায় হেত পাের- আমরা যিদ restriction িলেক relax কির তাহেল িক হেব। অথবা উলটাও িচ া করেত পােরা। আমরা এেক এেক restriction িলেক আেরাপ করেল িক হেব। সবার থেম আমােদর কােনা restriction নই। তাই আমােদর বর করেত হেব কয়িট divisor আেছ। এর একিট ফমু লা আেছ (*q*1 + 1)(*q*2 + 1)*...*(*qk* + 1) যখােন 2*N* = *pq*11 *pq*22 *...pqkk* . এখন যই িতনিট restriction আেছ তার মােঝ সবেচেয় কিঠন মেন হয় একিট জাড় আেরকিট িবেজাড়। িক এই ফমু লা অ সাের এিট খুব একটা কিঠন না। যিদ *p*1 = 2 হয় তাহেল (*q*1 + 1) এর টাম টা ফমু লা থেক সরােয় িদেত পারেলই হেব।

১সমান িক হেত পারেব? যেহতু একিট জাড় আেরকিট িবেজাড় হেত হেব সেহতু আসেল িট সমান কখনই হেত পারেব না।

৬৪

মােন আমরা 2 বােদ বািক prime িলেক সব ভােব distribute করিছ। এবার 2 িলেক একবার বাম িদেক আেরকবার ডান িদেক িদেলই হেব। উদাহরণ দয়া যাক, মেন কর 2*N* = 30. তাহেল 2 িল সিরেয় িনেল থােক 15. 15 এর divisor হে 1*,*3*,*5*,*15. এেদর সােথ 2 ন করেল িকছু divisor পােবা (ধরা যাক এেদরেক y িদেয় কাশ করা হয়) আর না করেল িকছু (ধরা যাক এেদর x িদেয় কাশ করা হয়)। িতিট y এর জ একিট x আেছ (x = 2N/y) আর উলেটাটাও সিত অথ াৎ িতিট x এর জ একিট y আেছ। এই x, y pair এর একিট ছাট হেব অপরিটর থেক আর সিটই হেব আমােদর *n*. কবল *n* = 1 হেত পারেব না এিটই বািক থাকল। এটা 1 িবেয়াগ কের িনেলই হেব। তেব *N* = 1 এর ে একটু সাবধান হেত হেব এই আর িক।

যমন 2*N* =30=2*×*3*×*5. তাহেল (1+1)(1+1) = 4. তরাং মাট 4 িট odd divisor. আর তােদর 2 িদেয় ন করেল আমরা আরও 4 িট পােবা। মাট 8 িট। এেদরেক *xy* আকাের িলখেল ধু ছাট িল n হেত পারেব। তরাং 8/2 = 4 িট। এই ছাট 4 িটর মােঝ একিট হেলা 1. তাহেল উ র 4 *−* 1=3. তরাং আমােদর উ র হেব (*q*2 + 1)*...*(*qk* + 1) *−* 1.

LOJ 1282 Leading and Trailing

**সমস া:** n এবং k দওয়া আেছ (2 *≤ n <* 231*,*1 *≤ k ≤* 107)। তামােক *nk* এর র 3 িডিজট এবং শেষর 3 িডিজট ি ট করেত হেব।

**সমাধান:** শেষর িতন িডিজট বর করা খুব সহজ। তামােক *nk* mod 1000 বর করেত হেব তাহেলই হেব (bigmod)। িক থম িতন িডিজট? এখােনই আেস log এর সৃজনশীল ব ব- হার। একটা উদাহরণ দখা যাক মেন কর 298751919. এেক log িনেল হেব log298751919 = 8751919log29 = 1*.*27987888233738 *×* 107. যিদও যেকােনা base এর log িনেলই হয় িক আমরা এখােন 10 base log িনেয়িছ। এই log এর মান দেখ িক বুঝা যায়? এভােব িচ া করেল মেন হয় বুঝা কিঠন হেয় যােব। একটু উলেটা ভােব িচ া করা যাক। ধরা যাক র িতনিট অংক হেলা *abc*. এর মােন আমরা আমােদর সংখ া *nk* ক িলখেত পাির 0*.abc×*10*d* যখােন d হেলা number of digit, আর আমরা ধু abc এই িতনিট র সংখ াই ধেরিছ। এর পেরও আরও অেনক িডিজট আেছ যটা আপাতত আমরা consider করিছ না, িক ব াপারটা মাথায় রােখা। এখন যিদ এর log নাও তাহেল হেব log(0*.abc ×* 10*d*) = log0*.abc*+*d*. 0 আর 1 এর মােঝর যেকােনা সংখ ােক log করেল negative সংখ া পাওয়া যায়। Negative সংখ া িনেয় িচ া করা একটু ঝােমলা। সজ- আমরা এই equation ক একটু ঘুিরেয় িলিখ- log *a.bc ×* 10*d−*1 = log *a.bc* + (*d −* 1). এখােন থম অংশ log*a.bc* যা অব ই ধনা ক কারণ *a.bc >* 1 (a হেলা *nk* এর থম িডিজট, আর থম িডিজট তা অব ই শূ হেব না তাই না?) আর *a.bc <* 10. তরাং আমরা বলেত পাির 0 *<* log*a.bc <* 1. তার মােন log*a.bc* + (*d −* 1) এর দশিমেকর আেগর সংখ া বেল *d−*1 এর মান আর তার পেরর সংখ া িল হেলা log *a.bc*. িকছু ণ আেগর উদাহরেণ আমরা দেখ- িছ log 298751919 = 1*.*27987888233738 *×* 107 = 12798788 + 0*.*8233738*...*. এ থেক আমরা বুঝেত পারিছ য 298751919 ত মাট 12798789 িট িডিজট আেছ। আর log*a.bc . . .* = 0*.*8233738*...*. তাহেল আমরা log*nk* হেত এর floor বাদ িদেলই আমরা log*a.bc . . .* পেয় যাব। িক আমােদর দরকার থম িতনিট িডিজট। সজ আমরা িদেক 10 এর power িনব, *a.bc . . .* = 100*.*8233738*...* = 6*.*6584605*...*. এর মােন আমােদর থম িতন িডিজট 665. িব- াস হে না? তামরা চাইেল wolframalpha.com বা python এ িহসাব কের দখেত পােরা 298751919 এর আসল মান কত। বা চাইেল একিট bigint এর কাড c++ বা java ত িলেখ ফলেত পােরা।

LOJ 1289 LCM from 1 to n

**সমস া:** মাট 10,000 িট case থাকেত পাের। তামােক n দওয়া আেছ 1 *≤ n ≤* 108, *lcm*(1*,*2*,...n*) বর করেত হেব। যেহতু উ র অেনক বড় হেব তাই উ রিট 232 এর mod এ িদেত হেব।

৬৫

**সমাধান:** উ র 232 এর mod এ িদেত হেব, তাই আমরা যিদ সব িহসাব unsigned int এ কির তাহেল mod আপনা আপিনই হেয় যােব।

আশা কির তামরা lcm বর করার নানা উপায় জান। িচ া কর এর মােঝ কান প িতিট এখােন কােজ দেব। LCM বর করার একিট উপায় হেলা িতিট াইম িনেয় দখা য তার সেব া কান power আমােদর দ সংখ ার কােনা না কােনা একিট সংখ ােক ভাগ কের। আমােদর সম ায় আমােদরেক 1 হেত n পয সব িল সংখ া দওয়া আেছ। আমরা যিদ 1 হেত n পয িতিট prime এর জ বর করেত পাির সই াইেমর সেব া কান power িট n এর সমান বা ছাট, তাহেলই আমরা lcm এ সই prime এর power পেয় যাব। যমন n যিদ 15 হয় আর আমরা যিদ 2 িনেয় মাথা ব াথা কির তাহেল 2 এর সেব া power যা খুব জাড় n সটা হেব 23 = 8, কারণ 24 = 16 িক 15 এর থেক বড়। তরাং আমােদর lcm এ 2 এর power হেব 3. িক িত case এ 108 পয সব prime িনেয় িনেয় এই চক করা খুবই সময় সােপ ব াপার। কারণ আমােদর case আেছ 10,000 িট। তাহেল উপায় িক?

মােঝ মােঝ খুব ভাল উপায় বর করার থেক বত মান প িতেক optimize করেত চাওয়া টা বুি মােনর মত কাজ হয়। যমন এখােন, একটু িচ া করেল বুঝেত পারবা য আসেল *√n* এর থেক বড় াইম িলর খুব জাড় 1 power থাকেব। তাই আমরা যিদ *√n* এর থেক বড় এবং সেব া *n* পয াইম িলর ণফল সহেজ এবং কম সমেয় বর করেত পারতাম তাহেল িত case এ *√n* সময় লাগত, যা ac পাবার জ যেথ । এই ে আমরা square root decomposition প িত ব বহার করেত পাির। 1 হেত 108 পয সকল াইম বর কের তােদরেক square root segment এ ভাগ কের িত segment এর ণফল বর কের রােখা। তাহেলই তামরা িতিট n এর জ খুব সহেজ *√n* হেত n পয prime িলর ণফল বর কের ফলেত পারেব। Square root decomposition না জানেল তা বই আর নট আেছই।

LOJ 1236 Pairs Forming LCM

**সমস া:** মাট 200 িট case থাকেত পাের। তামােক n দওয়া আেছ 1 *≤ n ≤* 1014, বলেত হেব কত িল i এবং j আেছ যন *lcm*(*i, j*) = *n* হয় যখােন 1 *≤ i ≤ j ≤ n*. **সমাধান:** মেন কর n ক আমরা prime factorization করলাম *pq*11 *pq*22 *...*. তাহেল িত *px* এর জ i এবং j ত *px* এর সেব া power হেত পারেব *qx*. ধু তাই না, i বা j এর কােনা একিটেত *qx* power থাকেতই হেব। যিদ এই শষ restriction িট না থাকত তাহেল i বা j ত power 0*,*1*,...qx* এর কােনা একিট হেত পারত। অথ াৎ i ত power হেত পারত *qx* + 1 রকম, j তও তাই। তরাং আমােদর মাট (*qx* + 1)(*qx* + 1) রকেমর power এর combination হেত পারত। িক শষ restriction বলেছ তােদর একিটেক *qx* হেতই হেব। ধরা যাক *i* ত এই *qx* power আেছ আর j ত নাই, তাহেল j এর power হেত পারেব 0*,*1*,...qx −*1 এই *qx* রকম, একই ভােব j ত *qx* power িক i ত না, তা হেত পাের *qx* ভােব। i ও j িটেতই power *qx* এিট হেত পাের 1 ভােব। তরাং

মাট 2*qx* + 1 ভােব এই power থাকেত পাের। আমরা যিদ িতিট prime এর power এর জ এই সংখ া বর কের ন করেত পাির তাহেল আমরা সকল i ও j এর সংখ া বর কের ফলেত পারব। তেব এখােন একটা সম া আেছ। আমােদর সম ায় বেলিছল *i ≤ j* হেত হেব। িক আমােদর এই counting তা মােন না। সিট মানেত হেল িক করেত হেব সটা মেন হয় তামরা িনেজরাই বর কের ফলেত পারেব। যিদ না পার, তাহেল কাগেজ কলেম বর কের দখ এই ই count অথ াৎ আমােদর আেলািচত count আর আসল উ র এর মােঝ পাথ ক কমন।

LOJ 1027 A Dangerous Maze

**সমস া:** একিট maze এ মাট *n* িট দরজা আেছ (সেব া 100 িট)। িতিট দরজার জ একিট সংখ া আেছ *ai*. যিদ *ai* ধনা ক হয় এর মােন হেলা এই দরজা িনব াচন করেল তুিম এই maze হেত *ai* সময় পর বর হেয় যােব। আর যিদ *ai* ঋণা ক হয় তাহেল তামােক *abs*(*ai*) সময় পর আবােরা এই maze এ ফরত আসেত হেব িক তখন ৃিত শি refresh কের দওয়া হেব, অথ াৎ

৬৬

এর আেগ কান কান দরজায় িগেয়িছেল তা তামার আর মেন থাকেব না। িতবার তুিম randomly দরজা িনব াচন করেব। তামােক বলেত হেব এই maze হেত ব েত expected কত সময় লাগেব। যিদ এই maze হেত বর হওয়া স ব না হয় তাহেল impossible ি ট কর।

**সমাধান:** যিদ কােনা দরজার জ ই *ai* ধনা ক না হয় তাহেল তা এই maze হেত বর হওয়া স ব না।

এখন মেন কর আমরা দরজা িলেক ই ভােগ ভাগ করলাম। এক ভােগ আেছ *ai* ধনা ক িল ধরা যাক এরা হেলা *x* (অথ াৎ দরজা িল হেব *x*1*,x*2 *...* এরকম)। আর অ ভােগ আেছ *ai* ঋণা ক দরজা িল যােদর আমরা *y* িদেয় িচনব। ধরা যাক maze হেত বর হেত আমােদর expected সময় লােগ *E*. আমরা randomly একিট দরজা পছ করলাম। কােনা একিট দরজা িনব াচন করার probability হেব 1/*n*. এখন যিদ আমরা *xi* দরজা পছ কির তাহেল এই maze হেত বর হেত সময় লাগেব *xi*. আর যিদ *yi* দরজা পছ কির তাহেল সময় লাগেব *yi* +*E*. কন? কারণ *yi* সময় পর আমরা maze এ িফের আসব, এর পর আমােদর expected সময় লাগেব E. তাহেল আমরা E সমান িলখেত পাির 1*n*(*x*1 + *x*2 + *...*) + 1*n*(*y*1 + *E* + *y*2 + *E* + *...*). যিদ এখেনা না বুেঝা তাহেল এভােব বুঝার চ া কর- আমরা randomly একিট দরজা িনব াচন করব। এিট *xi* হেল সময় লাগেব *xi* আর যিদ *yi* হয় তাহেল *yi* সময় তা লাগেবই সই সােথ *E* সময় লাগেব কারণ আমরা পুেরা েসস নতুন কের করিছ। তরাং িতিট দরজার জ কত সময় লাগেব তােক যিদ আমরা সই দরজা িনব াচন করার স াব তা িদেয় ন কের তােদর যাগ কির তাহেলই আমরা আমােদর expected value E পােবা। উপেরর সমীকরেণ E এর মান সমাধান করেলই আমরা আমােদর উ র পােবা।

এই সম ায় আিম ই া কেরই িকছু mathematically incorrect কথা বেলিছ যমন *xi* দরজা ত সময় লাগেব *xi*. িক আশা কির আমার মূল উে বুঝেত পারেব, কারণআিমযিদ দরজা বুঝােত অ কােনা variable ব বহার করতামবা mathematically correct ভােব বলেত চাইতাম তাহেল মেন হয় অেনক অেনক variable এবং definition আমােক বলেত হেতা।

LOJ 1030 Discovering Gold

**সমস া:** একিট হােত *n* িট ঘর (ঘর িল 1 হেত n ারা কাশ করা হয়) পাশাপািশ আেছ। হার িতিট ঘের িকছু সানা আেছ আর সই সানার পিরমাণ তামােক input এ বলা আেছ। তুিম েত 1 এ আছ। এই অব ায় তুিম একিট ছ া মারেব। ছ ায় 1 হেত 6 এর মােঝ কােনা একিট সংখ া randomly পড়েব, ধরা যাক i. তখন তুিম যিদ *x* তম ঘের থােকা তাহেল i পড়ার পর তুিম *x*+*i* তম ঘের যােব। যিদ *x* + *i>n* হয় তাহেল তুিম আবােরা ছ া মারেব। যত ণ না তুিম n এর ভতেরর একিট সংখ া পাও। যিদ তুিম n এ পৗঁছােয় যাও তাহেল তামার খলা শষ। তুিম যত িল ঘের যােব সব ঘর হেত সানা সং হ করেব। তামােক বলেত হেব এই খলায় n এ পৗঁছােল expected সানার পিরমাণ কত। n সেব া 100 হেত পারেব।

**সমাধান:** ধরা যাক *Ei* হেলা i তম ঘর হেত কের n তম ঘের িগেয় শষ করেল expected কত সানা পাওয়া যায় সই পিরমাণ ( সই ঘর সহ)। আমােদর বর করেত হেব *E*1 এর মান। মেন কর তুিম 1 এ থাকেত একিট ছ া মারেল। তাহেল এেত 1 হেত 6 এর যেকােনা একিট পড়েত পাের। ধরা যাক *x* পরল। তাহেল পেরর ঘর হেব 1 + *x*. এখান থেক তুিম expected *E*1+*x* িট সানা পােব। *x* পড়ার স া না 1/6. এরকম কের যিদ তুিম x এর মান 1 হেত 6 সব িল চ া কর তাহেল পােব *E*1 = *gold*[1] + 16(*E*2 + *E*3 + *...E*7) এরকম কের তুিম সব *Ei* এর মান বর কের ফলেত পারেব। সম া হেব যখন কােনা একিট *Ei* থাকেব না। যমন তুিম *En−*1 এ থেক যিদ 2 ফেলা তাহেল তুিম *En*+1 এ যেত চাইেব। িক *En*+1 বেল তা িকছু নই। সে ে িক করেব? সে ে তা তুিম ঐ ঘেরই থােকা তাই না? তার মােন তুিম আবােরা ছ া মারেব আর সে ে তামার expected মান হেব *En−*1 ই। অথ াৎ আমােদর equation হেব *En−*1 = *gold*[*n −* 1] + 16(*En* + *En−*1 + *En−*1 + *...*) ( মাট 5 িট *En−*1 থাকেব)। তরাং আমরা যিদ *i* = *n* হেত 1 পয *Ei* এর মান বর কির তাহেলই আমােদর সম া সমাধান হেয় যােব।

৬৭

আর এখােন base case হেব *En* = *gold*[*n*] কারণ n তম ঘের আসেল তুিম ধু ঐ ঘেরর সানা িনেয় খলা শষ কের দেব।

LOJ 1284 Lights inside 3D Grid

**সমস া:** একিট *X×Y ×Z* আকােরর ি ড দওয়া আেছ। ি েডর িতিট সেল একিট লাইট আেছ যা েত off. তুিম K বার একিট অপােরশন করেব। িত অপােরশেন তুিম ি েডর িট ঘর randomly িনব াচন করেব। ধরা যাক এরা হেলা (*x*1*,y*1*,z*1) এবং (*x*2*,y*2*,z*2). এবার তুিম *min*(*x*1*,x*2) *≤ x ≤ max*(*x*1*,x*2), *min*(*y*1*,y*2) *≤ y ≤ max*(*y*1*,y*2), *min*(*z*1*,z*2) *≤ z ≤ max*(*z*1*,z*2) এর জ সকল (*x,y,z*) এর লাইট িল toggle করেব। বলেত হেব K বার এই অপােরশন করার পর expected কত িল বা on থাকেব। X, Y, Z এর মান সেব া 100 আর K সেব া 10,000 হেত পাের।

**সমাধান:** Linearity ofExpectationবেল একিট কথা আেছ।এিটবেল অেনক িজিনেসর expectation হেব িতিট িজিনেসর expectation এর যাগফল। অথ াৎ যমন এই সম ােত, expected কত - িল লাইট on থাকেব সই পিরমাণ হেব িতিট লাইট expected কতবার on থাকেব তার যাগফল। অথ াৎ আমরা ি েডর িতিট লাইেটর জ বর করব সই লাইট K িট অপােরশেনর পর expected কতবার on থাকেব। িতিট লাইেটর জ এই সংখ া যিদ আমরা যাগ কির তাহেলই উ র পেয় যাব।ধরা যাক আমরা জানেত চাইিছ (*x,y,z*) K অপােরশন শেষ expected কত বার on থােক। এটা বর করা আর K বার শেষ on থাকার probability বর করা একই কথা। কন? উলেটা ভােব িচ া করেত পােরা। মেন কর K বার শেষ on থাকার probability q. তাহেল Expected কত বার on থাকেব? *q×*1+(1*−q*)*×*0 = *q*. তরাং আমরা K অপােরশন শেষ on থাকার probability

বর করেত চাইিছ।

ধরা যাক p হেলা ি েডর িট সল (*x*1*,y*1*,z*1) এবং (*x*2*,y*2*,z*2) িনব াচেনর probability যন (*x,y,z*) িব িট *min*(*x*1*,x*2) *≤ x ≤ max*(*x*1*,x*2), *min*(*y*1*,y*2) *≤ y ≤ max*(*y*1*,y*2), *min*(*z*1*,z*2) *≤ z ≤ max*(*z*1*,z*2) মেন চেল।আমরা p এর মান বর করব।একটা িজিনস খয়াল কর, আমরা িক x, yএবং z য যথা েম[*min*(*x*1*,x*2)*, max*(*x*1*,x*2)], [*min*(*y*1*,y*2)*, max*(*y*1*,y*2)], [*min*(*z*1*,z*2)*, max*(*z*1*,z*2)]এর ভতের থাকেব তার probability আলাদা আলাদা কের বর কের ন করেত পাির। তাহেলই আমােদর (*x,y,z*) িব িট আমােদর range িল মেন চলেব। তাহেল আমােদর এখন বর করেত হেব কত probability ত আমরা *x*1*,x*2 িনব াচন করেত পাির যন *min*(*x*1*,x*2) *≤ x ≤ max*(*x*1*,x*2) হয় দ x এর জ । এখােন "হয় না" বর করাটা সহজ হেব। "হয় না" হেব যিদ *x*1*,x*2 *< x* হয় অথবা *x<x*1*,x*2 হয়। x এর থেক ছাট মান আেছ *x −* 1 িট, আর বড় মান আেছ *X − x* িট। তরাং আমােদর িনব ািচত *x*1*,x*2 িটই ছাট হেব তার probability *x−*1 1*× x−*1 1আর িটই বড় হেব তার probability *X−x* 1*× X−x*1. এেদর যাগ কের 1 থেক িবেয়াগ করেলই আমরা x রে র ভতের থাকেব তার probability পেয় যাব। একই ভােব y এবং z ভতের থাকার probability বর কের সবাইেক ন করেলই আমরা p পেয় যাব।

এখন আমােদর বর করেত হেব এই p probability এর ঘটনািট K বােরর মােঝ িবেজাড় সংখ ক বার ঘটার probability কত। কারণ জাড় সংখ ক বার ঘটা মােন লাইট অফ থাকেব, আর িবেজাড় সংখ আমরা ক জািন বার ঘটা (*x* + মােন *y*)*n* লাইট = ∑অন *ni*=0

থাকেব। এবার লাগেব আমােদর binomial theorem এর ান।

(*ni*)*xiyn−i*. আবার (*−x* + *y*)*n* = ∑*ni*=0 আমরা এই িটেক যাগ কের পাই (*x* + *y*)*n* + (*−x* + *y*)*n* = ∑*i* 2(*ni*(*ni*)*xiyn−i* )যখােন (*−x*)*iy*i *n−i*. হেলা 0 হেত n এর মােঝর সকল জাড় সংখ া। তরাং আমরা যিদ x এর পিরবেত p আর y এর পিরবেত (1*−p*) বসাই তাহেল আমরা খুব সহেজ p probability এর ঘটনা জাড় সংখ ক বার ঘটার স াবনা পেয় যাব। তাহেল 1 হেত এই সংখ া বাদ িদেল িবেজাড় স কবার ঘটার probability ও পেয় যাব!আমার মেন হয় পুেরা সমাধােনর মূল অংশ েলা বলা হেয় িগেয়েছ। এখন পুেরাটু বুেঝ সমাধান কের ফলা তামার দািয় !

৬৮

৪.১ অ শীলনী

৪.১.১ সম া

SimpleX UVa 1648 Business Center X LOJ 1007 Mathematically Hard

X LOJ 1014 Ifter Party X LOJ 1028 Trailing Zeroes (I) X LOJ 1035 Intelligent Factorial Factorization X LOJ 1054 Efficient Pseudo Code X LOJ 1067 Combinations X LOJ 1077 How Many Points? X LOJ 1090 Trailing Zeroes (II) X LOJ 1098 A New Function X LOJ 1109 False Ordering X LOJ 1163 Bank Robbery

EasyX UVa 11040 Add bricks in the wall X UVa 1644 Prime Gap

X UVa 1210 Sum of Consecutive Prime Numbers X UVa 10539 Almost Prime Numbers X UVa 10622 Perfect Pth Powers X UVa 1646 Edge Case X UVa 10886 Standard Deviation X UVa 1647 Computer Transformation X UVa 11246 K-Multiple Free Set X UVa 11166 Power Signs X UVa 11303 Permutation X UVa 1655 Exam X LOJ 1024 Eid X LOJ 1138 Trailing Zeroes (III) X LOJ 1197 Help Hanzo X LOJ 1213 Fantasy of a Summation X LOJ 1214 Large Division X LOJ 1259 Goldbach`s Conjecture X LOJ 1333 Grid Coloring X LOJ 1336 Sigma Function X LOJ 1340 Story of Tomisu Ghost X LOJ 1370 Bi-shoe and Phi-shoe X LOJ 1038 Race to 1 Again X LOJ 1104 Birthday Paradox X LOJ 1151 Snakes and Ladders X LOJ 1248 Dice (III) X LOJ 1256 Word Puzzle X LOJ 1058 Parallelogram Counting

Medium

X UVa 808 Bee Breeding X UVa 294 Divisors X UVa 11526 H(n) X UVa 11105 Semi-prime H-numbers X UVa 1649 Binomial coefficients X UVa 10640 Planes around the World X UVa 10479 The Hendrie Sequence X UVa 766 Sum of powers X UVa 12520 Square Garden X UVa 12590 Guards II X UVa 1652 Fibonacci System X UVa 11895 Honorary Tickets X UVa 10976 Fractions Again?! X LOJ 1234 Harmonic Number X LOJ 1298 One Theorem, One Year X LOJ 1318 Strange Game X LOJ 1161 Extreme GCD

৬৯

৭০

অধ ায় ৫

Bruteforce এবং Backtrack স িক ত সম া

Bruteforce এবং backtrack কােনা রকেমর smart algorithm না। বরং এিট গাধার মত সাজা সাপটা ভােব যা করেত বেলেছ তা করা। িক তাই বেল বিশ গাধার মত করেল হেব না। হয়েতা মােঝ মােঝ তামােক ছাট খােটা ি খািটেয় সমাধানিট সহেজ কাড করেত হেব। বা কােনা সময় ছাট খােটা optimization কের run time ক কিমেয় আনেত হেব।

UVa 725 Division

**সমস া:** N দওয়া থাকেব (2 *≤ N ≤* 79). তামােক িট 5 িডিজেটর সংখ া abcde ও fghij বর করেত হেব যােত কের *abcde fghij* = *N* হয় এবং এই 5 িডিজেটর িট সংখ ার 10 িট িডিজট আলাদা আলাদা হয়। যমন *N* = 62 এর জ একিট উ র হেত পাের 79546

01283.

**সমাধান:** যেহতু fghij একিট 5 িডিজেটর সংখ া আমরা এর জ 0 হেত 99999 পয লুপ চালােত পাির। এরপর এেক input এর N এর সােথ ন করেল abcde বর হেয় যােব। আমরা এরপর চক কের দখেত পাির য এই িট সংখ ার সকল িডিজট আলাদা আলাদা িকনা। যিদ হয় তাহেল এই জুিট আমােদর একিট সমাধান।

তামরা চাইেল িকছু ছাট খােটা optimization করেত পােরা। যমন fghij এর লুপ 0 হেত না চািলেয় 1000 হেত চালােনা। কারণ এর থেক ছাট িট সংখ ার থম িট িডিজট শূ । আবার যিদ দেখা য N ক fghij িদেয় ন করেল সংখ ািট 99999 এর থেক বড় বা 1000 থেক ছাট হয় তাহেল আর ক কের চক করার দরকার নই। এই সম ােত এই ধরেনর সহজ optimization দরকার নই, তেব হয়েতা অ কােনা সম ায় দরকার হেত পাের।

UVa 11059 Maximum Product

**সমস া:** তামােক n িট সংখ ার একিট অ াের *S*1*,S*2*,...Sn* দওয়া থাকেব যখােন 1 *≤ n ≤* 18 এবং *| Si |≤* 10. তামােক এর একিট continuous subarray বর করেত হেব যন তােদর

ণফল সেব া হয়।

**সমাধান:** n মা 18, তাই আমরা চাইেল এর সকল sub array িনব াচন কের তার element িলেক ন করেত পাির। তেব একটা িজিনস, যেহতু অেনক িল সংখ া ন করিছ আমােদর খয়াল রাখেত হেব ণফল যােত overflow না কের। আমােদর সংখ া িল খুব জাড় 10 হয়। 18 িট সংখ ার ণফল

৭১

তাই সেব া 1018 হেত পাের যা long long এ ধরেব। তরাং আমােদর long long এর variable িনেয় িহসাব িনকাশ করেত হেব। int িনেয় কাজ করেল হেব না, overflow হেব।

UVa 524 Prime Ring Problem

**সমস া:** তামােক একিট জাড় সংখ া n দওয়া থাকেব (1 *≤ n ≤* 16). একিট সাইেকেল 1 হেত n পয সংখ া িলেক কত ভােব সাজােনা যায় যন পাশাপািশ যেকােনা িট সংখ ার যাগফল একিট prime হয়। যমন *n* = 6 এর জ 143256 একিট সমাধান।

**সমাধান:** আমরা সাইেকেলর েত 1 বসাই। এরপর আমােদর কাজ হেলা 2 হেত n পয *n−*1 িট সংখ া এেক এেক সাইেকেল বসােনা। আমরা এিট (*n −* 1)! ভােব করেত পাির। িক *n* = 16 এর জ এিট হেব 15! যা একটু বিশই বড় হেয় যায়। এখােন একটা িজিনস ল িনয়। পাশাপািশ িট সংখ া বসােল prime পাওয়া যােব এরকম ঘটনা খুব একটা ঘটেব না। তাই আমরা যা করেত পাির তাহেলা সাইেকেল আমরা যখন সংখ া বসাব তখনই নতুন সংখ া আর তার পােশর সংখ া যাগ কের দখেত পাির য তােদর যাগফল াইম হয় িকনা। না হেল আর এ েনার দরকার নই। আর হেল পেরর জায়গা িনেয় আমরা মাথা ব াথা করব। অথ াৎ backtrack এর মত কের সাইেকেল আমরা এেক এেক সংখ া বসােত থাকব। মেন কর আমােদর backtrack এর ফাংশন হল *bktk*(*at*). এর মােন আমরা এখন at তম ােন সংখ া বসােনার চ া করব। আমরা একিট অ াের রাখব যখােন 0-1 ব বহার কের িলেখ রাখব কান সংখ া ব বহার করা হেয়েছ আর কান সংখ া ব বহার করা হয় নাই। তরাং আমরা at এ এেস সই অ ােরর উপর িদেয় লুপ চািলেয় দখব য কান কান সংখ া ব বহার করা হয় নাই। তােদরেক at এ বসােনার চ া করব। চ া করা মােন দখব য এই সংখ া এর আেগর সংখ ার সােথ যাগ করেল যাগফল াইম হয় িকনা। না হেল তা আবার আমরা পেরর সংখ া িনেয় চ া করব। আর যিদ াইম হয় তাহেল অ ােরেত িলেখ রাখব য এই সংখ া ব বহার করা হেয় িগেয়েছ এবং *bktk*(*at* + 1) কল করব। একটা িজিনস তা হল, *bktk*(*at* + 1) এর পেরর লাইেন, অথ াৎ *bktk*(*at*+1) হেত িফের এেস তামােক যই সংখ া ব বহার কেরছ সই সংখ ােক অ ােরেত ব বহার করা হয় নাই বলেত হেব, অথ াৎ *bktk*(*at* + 1) কল করার আেগ যিদ আমরা *used*[*i*]=1 কের থািক তাহেল ফাংশন হেত িফের আমােদরেক *used*[*i*]=0 করেত হেব। আর শষ সংখ া বসােনার সময় তার সােথ র সংখ া যাগ কের দখেত ভুেল যও না।

UVa 12325 Zombie's Treasure Chest

**সমস া:** একিট সানা দানার দাকােন িগেয় দখেল সখােন ই ধরেনর র আেছ। থম ধরেনর রে র িতিটর ওজন *s*1 এবং িতিটর মূল *v*1. ি তীয় ধরেনর রে র ওজন *s*2 এবং মূল *v*2. িত ধরেনর র অসংখ সংখ ক আেছ। তামার কােছ মাট n ওজেনর র নওয়া যােব এরকম একিট থেল আেছ। তামােক বলেত হেব সবেচেয় কত বিশ মূেল র র তামার থেল ত িনেত পারেব। *n, s*1*,s*2*,v*1*,v*2 িতিট 32 bit এর integer. **সমাধান:** এর সমাধানিট আমার কােছ বশ interesting লেগেছ। ধরা যাক *s*1 *> s*2. এখন িট case ক না কর। থম case, *s*1 *> √n*. এই ে থম িজিনস তুিম *n*/*s*1 এর বিশ বার িনেত পারেব না আর যেহতু *s*1 *> √n* তাই তুিম আসেল *√n* এর বিশ বার থম র িনেত পারেব না। কারণ *s*1 ওজেনর িজিনস তুিম যিদ *√n* এর থেক বিশ বার যিদ নাও তাহেল এই িজিনস িলর ওজন হেব *s*1*√n* এর বিশ। যেহতু *s*1 *> √n* সেহতু মাট িজিনেসর ওজন *n* এর থেক বিশ হেয় যায়। িক তা তা স ব না। তরাং তুিম খুব সহেজই থম র কয়িট নেব তার উপর লুপ চালােত পারেব (সেব া *√n* িট িনেত পারেব)। ধরা যাক i িট নেব। তাহেল বািক থাকেব *x* = *n − is*1 পিরমাণ জায়গা। এই জায়গায় তুিম ি তীয় র সেব া *x*/*s*2 িট িনেত পারেব।

সম া হেলা যখন *s*1 *≤ √n* হেব। এর মােন *s*2 ও িক ছাট। এই ে একিট িবধা আেছ। *s*2 সংখ ক থম রে র ওজন আর *s*1 সংখ ক ি তীয় রে র ওজন সমান (উভয়ই *s*1*s*2)। তুিম কানিট নেব? যার িত কিজ মুল বিশ অথ াৎ যার *vi*/*si* বিশ তাই না? ধরা যাক *v*1/*s*1 *> v*2/*s*2. এর

৭২

মােন তুিম চ া করেব থম ধরেনর র বিশ বিশ কের িনেত। অথ াৎ তুিম ি তীয় র িট কখনই *s*1 এর সমান বা এর থেক বিশ সংখ ক নেব না। িনেল সই *s*1 িট ি তীয় রে র পিরবেত *s*2 িট থম র িনেত পারেব। যেহতু *s*1 *≤ √n* তরাং তুিম ি তীয় র 0 হেত *s*1 *−* 1 েত ক ভােব নবার চ া করেব। আর বািক অংশটু থম র িদেয় ভরােনার চ া করেব। এভােব সকল কার ভােব চ া করেল তুিম optimal উপায় পেয় যােব।

UVa 1343 The Rotation Game

**সমস া:** এই সম ায় িচ ৫.১ এর মত একিট বাড দওয়া থাকেব। বাড িটেত 1, 2 এবং 3 সংখ া- িল মাট 8 বার কের থাকেব ( তরাং মাট 24 িট সংখ া)। তুিম দ বােড 8 রকেমর অপােরশন করেত পারেব। এ িল িচে *A,B...H* িদেয় কাশ করা হেয়েছ। অথ াৎ িতিট অপােরশন হেলা বড় row বা column ক cyclically উপের-িনেচ বা ডােন-বােম ঘুরান। যমন িচে র থম বাড েক A বরাবর ঘুরােল ি তীয় বাড পাওয়া যায়, আবার ি তীয় বাড েক C বরাবর ঘুরােল তৃতীয় বাড পাওয়া যায়। তামােক ঘুিরেয় ঘুিরেয় এমন অব ােন আনেত হেব যন মােঝর 8 িট ঘের একই সংখ া থােক। যমন িতন না ার বােড মােঝর 8 িট ঘের 2 আেছ। তামােক এই কাজ সবেচেয় কম সং- খ ক অপােরশন ব বহার কের করেত হেব। এরকম সমাধান একািধক থাকেল lexicographically smallest উ রিট িদেত হেব।

নকশা ৫.১: UVa 1343 এর বাড

**সমাধান:** িন য় এসম া নানা ভােব সমাধান করা যায়। আমরা এখােন IDA\* ব বহার করব। IDA\* এর পূণ প হেলা iterative deepening A\*. ভ াগ বশত আিম আমার " া ািমং কে ট " বইেয় IDA\* স েক িকছুই িলিখ নাই। তাই এখােন এ িনেয় িকছু িলখিছ।

IDA\* বুঝার আেগ আমােদর বুঝেত হেব searching িক। মেন কর আমােদর এই সম ায় র ান (state) একিট বাড । এর সােথ িকছু অপােরশন দওয়া থাকেব। কােনা এক ােন যিদ কােনা অপােরশন কর তাহেল তুিম আেরক ােন যােব। আমােদর ল হেলা সবেচেয় কম অপােরশন ব বহার কের আমােদর কাি ত লে পৗঁছান। এিটই হেলা searching. অথ াৎ দ অপােরশন সমূহ ব বহার কের র ান হেত লে সবেচেয় কম খরেচ পৗঁছানই হেলা searching (আসেল সবেচেয় কম খরেচ পৗঁছান সবসময় ল থােক না। িক আমােদর আেলাচনার িবধার জ আমরা ধের নই সবেচেয় কম খরেচ পৗঁছােত চাই)।

যিদ উপেরর প ারা ভাল কের পের থােকা তাহেল দখেব এখােন িতনিট িজিনস আেছ। এক- র ান, ই- ল , িতন- অপােরশন। একটা িজিনস, ল িক একটা িনিদ ান নাও হেত পাের। এটা হে একিট rule যা িদেয় তুিম বুঝেত পারেব য তুিম তামার লে পৗঁেছ িগেয়ছ। যমন এই সম ােত ল িক নানা রকম হেত পাের। তেব ল বুঝার rule হেলা মােঝর 8 িট ঘের একই সংখ া থাকা। অ ঘর িলেত যা খুিশ থা ক তা িনেয় মাথা ব াথা নই।

তাহেল এই searching সম া সমাধান করবা কীভােব। একিট উপায় হেত পাের bfs বা dfs কের সমাধান করা। িক আমরা dfs কের সমাধান করেত পারব না। কন? মেন কর A হেত B ত যাওয়া যায় (এখােন A, B, C ইত ািদেক state িহসােব িচ া কর, এই সম ার সােথ এেদর কােনা স ক নই)। আবার A হেত C হেয় B ত যাওয়া যায়। এখন dfs তা চাইেল A-C-B এই রা ােত

৭৩

B ত যেত পাের ফেল A-B য ছাট রা া আেছ সটা আিব ার হেব না। তরাং dfs িদেয় আসেল সবেচেয় ছাট রা া বর করা স ব হয় না। তাহেল িক আমরা bfs িদেয় সমাধান করেত পাির? হয়েতা হেব। িক সম া হেলা আমােদর state সমূহ save কের রাখেত হেব তােদর visited মাক করেত হেব। এসব করেত হয়েতা অেনক সময় লাগেব ( ােফ সাচ করেত সময় লাগেব না, সময় লাগেব state মাক ও save করেত) এবং অেনক মমিরও লাগেব।

এই সম ােক দূর করার জ আেছ IDA\* search. IDA\* search এ তমন কােনা মমির লােগ না, state ও মাক করেত হয় না। িবিনমেয় search করেত একটু বিশ সময় লােগ এটাই সম া। তামরা যিদ এখেনা backtracking িনেয় একদমই না পেড় থােকা তাহেল তামরা একটু পেড় আেসা। চাইেল নট বা আমার আেগর বইেয় দখেত পােরা।

IDA\* বলার আেগআিম তামােদর ID-DFSিনেয় বলব। এর পূণ পহেলাiterativedeepening depth first search. মেন কর আমােদর াফিট unweighted. আমােদর dfs এর বািহের একিট depthLimit এর লুপ থাকেব। থেম এই depthLimit এর মান হেব 1. আমরা যটা করব তাহল একটা dfs চালাব যার depth এর limit হেব depthLimit. অথ াৎ dfs করার সময় আমরা এর বিশ depth এ যেত পারব না। আরও িব ািরত বলেল বলা যায়, সাধারণত আমােদর dfs ফাংশেনর parameter এ থােক x অথ াৎ বত মান নাড। এর সােথ সােথ এখন আরও িট parameter থাকেব, current depth এবং max depth. িতবার dfs কল করার সময় আমরা current depth এর মান এক কের বািড়েয় দব। আর যিদ কখেনা দিখ সই মান max depth এর থেকও বড় হেয় িগ- য়েছ তাহেল আর না এিগেয় ফরত আসব। এর ফেল লাভ িক হে ? লাভ হে য আমরা source নাড হেত target এ shortest path এ যেত পারব। কন? কারণ থেম আমরা 1 max depth িদেয় চ া করব, এই ে যিদ dfs আমােদর target এ পৗঁছায় তাহেল আমরা জানব 1 উ র। যিদ তা না হয় তাহেল max depth ক বািড়েয় 2 করব এবং আবােরা dfs কল করব, এবার target এ পৗঁছােত পারেল 2 উ র হেব। না হেল আবার বািড়েয় 3 কের dfs. এভােব চলেত থাকেব যত ণ না আমরা target পাি । খয়াল কর এই dfs এ আমরা visited বেল িকছু রাখিছ না বা রাখার দরকার নই। তাহেল এভােব ID-DFS এর মাধ েম আমরা source হেত target এর shortest path বর কের ফলেত পারব। সম া িক তা তা বুঝেতই পারছ, একই state বার বার িভিজট করিছ, সময়

বিশ লাগেছ।

এখন চল আমরা আমােদর সম ায় একটু সমেয়র জ ফরত যাই। আমরা চাইেল আমােদর সম াও ID-DFS এর মাধ েম সমাধান করেত পাির। যিদও তােত অেনক বিশ সময় লাগেব। িক সমাধান তা হেব। হেলা আমরা আমােদর সমাধানেক কীভােব improve করেত পাির। উ র- heuristic. এটা আবার িক? এিট এমন একিট ফাংশন যােক যিদ তুিম একিট state দাও তাহেল এিট তামােক বলেব এই state হেত সমাধােন পৗঁছােত কমপে আরও কত move লাগেব। এই "আরও কত move লাগেব" এই িজিনেসর যই heuristic যত ভাল approximation িদেত পারেব সই heuristic তত ভাল। যমন আমােদর সম ার একিট heuristic বর করা যাক। আমােদর সম ায় মােঝর 8 িট ঘেরর মােঝ ধরা যাক x িটেত 1 নই। তাহেল আমরা বলেত পাির অ ত x move লাগেব সব িলেত 1 হেত। এরকম কের আমরা বর করেত পাির সব িল 2 এবং সব িল 3 হেত অ ত কত cost লাগেব। এই িতনিট সংখ ার মােঝ সবেচেয় কমিটই আমােদর heuristic এর মান হেত পাের। কন? কারণ এক move িদেল মােঝর একিট মা ঘেরই তামার কাি ত সংখ া আসেত পাের তাই না? (ব াপারটা ওত সহজ না, তাই একটু িনেজ িনেজ িচ া কের দেখা)

এখন হেলা আমরা কীভােব এই heuristic ব বহার করেত পাির। আমরা আমােদর ID- DFS এ িফের যাই। যিদ আমােদর কােছ এরকম কােনা heuristic থােক তাহেল আমরা িকছু optimization করেত পাির। মেন কর আমরা আিছ current depth এ আর আমােদর depth এর সেব া িলিমট হেলা max depth. এখন আমরা আমােদর state এর heuristic মান বর করব ধরা যাক h. যিদ দিখ current depth আর h এর যাগফল max depth ক ছািড়েয় িগেয়েছ তাহেল তা আর এই পেথ িগেয় লাভ নই। তাই dfs আর না এিগেয় ফরত যাব। এখােন িক হেয়েছ খয়াল কেরছ? তুিম জােনা বত মান state এ আসেত তামার কত ধাপ লেগেছ। আর তুিম heuristic থেক বর কেরছ য আর কত ধাপ অব ই লাগেবই (heuristic সবসময় তামােক "at least" মান দয়)। এ থেক তুিম খুব সহেজই বর করেত পােরা য এই রা া ধের গেল তামার target এ পৗঁছােত কত ধাপ লাগেবই। এখন যিদ দেখা এই ধাপ সংখ া তামার max depth কও ছািড়েয়

৭৪

যায় তাহেল তা আর এই পেথ িগেয় লাভ নই তাই না? এটাই IDA\*১। তুিম চাইেল িকছু ণ আেগ বলা heuristic ক কােজ লািগেয় আমােদর এই সম ািট সমাধান কের ফলেত পােরা িঠক আিম

যভােব বললাম সভােব। খুব সহজ কাড।

িকছু ণ আেগ বললাম িঠক এটাই IDA\* না আরও িকছু আেছ। এই আরও িকছু িক? তুিম চাইেল একটু িচ া করেত পােরা আমােদর সমাধােন আর কাথায় উ িত করা যায়। খুব সহজ উ িত িক । মেন কর আমােদর দওয়া max depth এর মােঝ সমাধান নই। তাহেল আমরা িক করিছলাম? max depth ক এক বাড়াি লাম। িক আমরা চাইেল এক না বািড়েয় একটু বিশ বাড়ােত পাির। আমরা তা অেনক অেনক নাড হেত current depth আর h এর যাগফল দেখ ফরত এেসিছ তাই না? আমরা িক চাইেল এেদর মােঝ সবেচেয় ছাটিটেক আমােদর পরবত ী max depth িহসােব set করেত পাির। অথ াৎ, আমরা max depth এ িলিমট থাকা অব ায় যই সব নাড হেত ফরত এেসিছ সইসব নাড হেত আমরা target এ পৗঁছানর একটা estimate পেয়িছলাম। সই সব estimate এর মােঝ সবেচেয় ছাটিট আমরা পেরর max depth িহসােব িনেত পাির। কারণ অ কােনা উপােয় তা এর থেক কেম target এ যাওয়া স ব না তাই না? তাই আমরা চাখ ব কের max depth না বািড়েয়, এরকম বুি কের max depth ক যিদ বাড়াই তাহেলই আমােদর algorithm িট IDA\* হেয় যােব। তামরা চাইেল wiki ত IDA\* এর pseudocode দেখ িনেত পােরা বুঝার িবধার জ ।

UVa 1374 Power Calculus

**সমস া:** মেন কর তামােক x দওয়া আেছ, তামােক *xn* বর করেত হেব। এই কাজ তুিম নানা ভােব করেত পােরা। যমন *n* = 7 এর ে , তুিম থেম *x*2 বর করেত পােরা x ক x এর সােথ ন কের। *x*2 ক *x*2 এর সােথই ন কের *x*4 বর করেত পােরা। এর পর এেক *x*2 এবং সবেশেষ x এর সােথ ন কের *x*7 বর করেত পােরা। এর ফেল মাট 4 িট ন করা লাগেলা। তা না কের তুিম চাইেল, *x*4 বর করার পর আবােরা এেক িনেজর সােথ ন কের *x*8 বর করেত পােরা আর এেক x িদেয় ভাগ কের *x*7 বর করেত পােরা। এে ে ও তামার 4 িট ন-ভাগ করেত হে । তামােক n দওয়া থাকেব (1 *≤ n ≤* 1000). বলেত হেব সবেচেয় কম কতিট ন ভাগ ব বহার কের তুিম *xn* বর করেত পারেব। **সমাধান:** থম কথা এিট dp না। কারণ তুিম এখন *xa* ত আছ এটাই ধু পূণ নয়। এিট তুিম কীভােব পেয়ছ, আর িক িক power তুিম জােনা সটাও পূণ । যমন মেন কর তুিম *x*100 বর করার পেথ যিদ *x*25 বর কের থােকা তাহেল তুিম একবার ভাগ কেরই *x*75 বর করেত পারেব। িক তুিম যিদ *x*25 না বর কের থােকা, তাহেল হয়েতা একটু কিঠন হেয় যােব *x*75 বর করার জ ।

এই সম া সমাধােনর জ তামােক backtrack করেত হেব। এজ আমােদর িকছু optimization এবং heuristic বর করেত হেব। মেন কর এখন পয তুিম *xa* বর কেরছ সবেচেয় বড়। আর তুিম *xn* চাও। যিদ কউ িজ াসা কের আর কত move লাগেবই। তুিম িক করেব? একটা উপায় হেলা এক move পর খুব জাড় *x*2*a* পােব, ই move পর *x*4*a* এরকম কের b move পর খুব জাড় *x*2*ba* পাবাই। এখন যিদ িহসাব কের দেখা *xn* পেত হেল কমপে আর কত move লাগেব অথ াৎ সবেচেয় ছাট কান b এর জ 2*ba ≥ n* হেব তাহেল সই মানেকই তামরা heuristic িহসােব ধরেত পােরা।

আেরকিট বশ পূণ optimization আেছ। আর তাহেলা সবসময় আেগর move এ যই *xa* পাওয়া িগেয়েছ তােক ব বহার কের সমাধান পাওয়া যােব। অথ াৎ আেগর move এ তুিম *x*10 পেয়ছ, তুিম এই move এ সিট ব বহার করেব। এটা কমন জািন। এর মাণ িক? এমনও তা হেত পাের য অ িট power ক ন ভাগ কের যা পাওয়া যায় তার সােথ পের হয়েতা *x*10 ক ন ভাগ করেত হেব। হেত পাের। তাহেল িক করা যায়? একিট উপায় হল বেস বেস মাণ করা য আসেলই এই optimization সিঠক। অথবা তুিম 1 হেত 1000 পয এই optimization সহ এবং বােদ ই ভােবই উ র বর কের দেখা। যিদ দেখা য সব ে ই উ র একই আেস, তার মােন ধের িনেত পােরা য তামার optimization িঠক আেছ।

১উম... িঠক এটাই না, আরও একটু আেছ, পের বলিছ

৭৫

মেন হয় এই ই optimization / heuristic ব বহার কের IDA\* বা backtrack করেল accepted হেয় যােব।

UVa 1602 Lattice Animals

**সমস া:** তামরা িক tetris খেলছ? যিদ tetris খেল না থােকা তাহেল এই সম ািট ছিব ছাড়া কম কথায় বুঝােনা একটু কিঠন হেয় যােব। তাই তামরা সাইেট পেড় িনেত পােরা। n সাইেজর polymino হল n িট বগ পাশাপািশ বিসেয় একিট connected িজিনস বানান। আমরা িট polymino ক একই বলব যিদ একিট থেক আেরকিট ঘুিরেয় বা flip (mirror) কের বা translate কের পাওয়া যায়। যমন *n* = 3 এর জ ই রকম polymino পাওয়া যায়। িতনিট বগ পাশাপািশ, অথবা L এর মত কের িতনিট বগ লাগান। আর অ িল হয় rotate কের বা flip কের বানােনা স ব। তামােক n, w এবং h দওয়া আেছ বলেত হেব *w × h* সাইেজর একিট grid এ কত িল n সাইেজর িভ polymino পাওয়া স ব। (1 *≤ n ≤* 10*,*1 *≤ w, h ≤ n*)

**সমাধান:** আেগর িট সম া হেত এই সম ার মুল পাথ ক হেলা আেগর িট সম ায় একিট সমাধান বর করেত বলা হেয়েছ বা shortest path এ সমাধান বর করেত বলা হেয়েছ। িক এখােন মাট কত িল সমাধান আেছ তা বর করেত বেলেছ। তরাং এে ে IDA\* বা BFS/DFS কােজ আসেব না। হয় আমােদর dp বা mathematically counting কের বর করেত হেব, নাহেল backtrack কের বর করেত হেব। এে ে আসেলmathematical ভােব counting করা মেন হয় না খুব একটা সহজ হেব। আর তাছাড়া n এর সাইজও কম আেছ। তাই এখােন backtrack বা simulation জাতীয় প িতই উপায়।

আমােদর সমাধােন n িট িল থাকেব।i তমিলে থাকেব i সাইেজর সকলিভ িভ polymino. িভ মােন এেদর কাউেক flip, rotate বা translate করেল এই িলে র অ কােনা polymino পাওয়া যােব না। এইসব polymino এর সােথ আরও একিট cell সকল ভােব জুেড় িদেয় i+1 সা- ইেজর polymino িল পাওয়া যােব। িক দখা যােব এেদর একিটেক translate, rotate বা flip কের অ আেরকটা পাওয়া যায়। তাই আমােদরেক i+1 হেত i+2 সাইেজর polymino িল বর করার আেগই i+1 এর িল হেত এসব similar polymino দর দূর করেত হেব। উদাহরন প মেন কর আমােদর কােছ একিট polymino আেছ (2*,*2)*,*(2*,*3). এখন আমরা একিট কের নতুন cell লািগেয় এেক (2*,*2)*,*(2*,*3)*,*(2*,*4) বানােত পাির, আবার (2*,*1)*,*(2*,*2)*,*(2*,*3) ও বানােত পা- ির। িক এই িট িক আবার একই (translation সােপে )। এেদর জনেকই রেখ আমােদর লাভ নই। তরাং আমরা যিদ একিট polymino এর িল থেক ধু িভ িভ polymino দর রাখেত পাির তাহেল আমরা 1 সাইজ িদেয় কের এেক এেক 2, 3, 4 এরকম সকল সাইেজর জ list পেয় যাব। হেলা কীভােব আমরা এই িভ িভ polymino িল পেত পাির বা িট polymino য একই তা কীভােব বুঝেত পাির।

এজ আমােদর যা করেত হেব তাহল polymino দরেক canonical ফেম কাশ করেত হেব। Canonical ফম মােন হল িতিট polymino ক এমন ভােব কাশ করা যন একই রকম (এে ে flip, translate আর rotate এর সােপে ) polymino দরেক canonical ফেম কাশ করেল একই canonical ফম নয়। এেত লাভ িক? লাভ হল কােনা একিট polymino এর িলে সকল polymino এর জ যিদ আমরা canonical ফম বর কির তাহেল িভ িভ canonical ফম ওয়ালা polymino িনেলই আমােদর িভ polymino এর িল হেয় যােব। এখন হল একিট polymino ক কীভােব canonical ফেম represent করা যায়। একিট polymino ক আমরা square িলর coordinate স িলত একিট set িহসােব represent করেত পাির। িক এিট িক canonical form না। কারণ িচ া কর 1 সাইেজর একিট polymino (0*,*0) হেত পাের অথবা (1*,*1) হেত পাের বা অ কােনা িকছু। িক এই সব িলই একই (translation সােপে )। তাহেল এেদরেক canonical ফেম কীভােব নয়া যায়?

যােত translation এর ফেল একািধক polymino ক িভ িবেবচনা না কির সজ একিট সহজ উপায় হেলা polymino এর bounding rectangle এর একিট িনিদ কাণা (ধরা যাক িনেচর বাম কাণা) ক িনেয় (0*,*0) ত বসােত হেব। এজ তামােক square িলর সবেচেয় কম x িনেত হেব এবং সবেচেয় কম y. এরপর সব square হেত তুিম যিদ এই (*x, y*) িবেয়াগ কর তাহেলই

৭৬

canonical form পেয় যাবা যা তামােক translation এর হাত হেত র া করেব। উদাহরন- (0*,*0) আর (1*,*2) এই জেনর ে তুিম যিদ উপেরর কাজ কর তাহেল িক ই ে ই (0*,*0) পােব, এর মােন আমরা জনেক রাখব না, এেদর এক জনেক রাখেলই চেল। Rotation বা flip এর জ িক করা যায়? এে ে ও নানা উপায় আেছ। একিট উপায় হেলা polymino ক rotate কর (4 ভােব rotate করা যায়, 0 বার, 1 বার, 2 বার আর 3 বার 90 িড ী rotate করা), flip করা ( ধু যেকােনা একিট flip িদেলই চলেব horizontal বার vertical), flip করার পর rotate করা (আবােরা 4 ভােব rotate)- এভােব যত িল ( মাট 8 রকম) polymino পােব সব িলই তা আমােদর সম া মেত একই তাই না? এেদর েত েকর translation সােপে canonical form

বর কর আর তােদর একিট সেট রাখ। খয়াল কর একিট polymino বা তার translation সােপে canonical ফম িক িনেজই একিট cell এর সট। আমরা এই polymino িলর জ যিদ একিট

সট নই তাহেল এিট হেব cell এর সেটর সট।

আমরা তাহেল িক করিছ? আমরা একিট িলে র েত ক polymino িনি , িনেয় তােদর িবিভ ভােব rotate আর flip কের তােদর translation সােপে canonical ফম বর করিছ। এর পর সই canonical ফম েক একিট সেট পুের িদি । এই কাজ সব polymino এর জ কের ফলেল আমরা আমােদর সেট িক সকল িভ িভ polymino পেয় যাব।

তাহেলআমােদর সমাধান ায় শষ, ধু একিটিজিনস বািক আেছ- কীভােবনতুন নতুন polymino পাওয়া যােব। আমরা ইেতামেধ ই 1 সাইেজর সকল polymino পেয় িগেয়িছ। এখন আমরা চ া করব i সাইেজর polymino িল হেত *i*+1 সাইেজর polymino বর করার। যিদ তা স ব হয় তা- হেল আমরা 1*,*2*,...n* সাইেজর সকল polymino বর কের ফলেত পারব। i সাইেজর polymino হেত *i*+1 সাইেজর polymino বর করার উপায় হেলা i সাইেজর polymino এর িতিট square এর চািরিদেক নতুন square (*i* + 1 তম, কারণ ইেতামেধ ই i িট square আেছ) বসােনার চ া করা। তাহেল আমরা একিট একিট কের নতুন polymino পােবা *i* + 1 সাইেজর। চ া করার কথা বললাম কারণ এমনও তা হেত পাের য যখােন বসােত চা সখােন আেগ থেকই একিট square আেছ। তরাং সই ব াপাের একটু সাবধান হেত হেব। তাহেল এভােব আমরা *i* + 1 সাইেজর জ একিট িল পেয় যাি । এরপর এেদর থেক আমােদর একই রকেমর polymino িল বাদ িদেত হেব (উপের canonical ফম িনেয় অেনক বকবক কেরিছ)। এভােব আমরা এেক এেক n সাইেজর িল পেয় যাব।

আর *w × h* এটা কােনা ব াপার না। একবার সব polymino পেয় গেল তুিম সব *w × h* এর জ উ র বর কের রাখেত পারেব। অথ াৎ তামার কাড একবার মা n সাইেজর সব polymino বর করেব এরপর তুিম একিট *n × w × h* সাইেজর একিট matrix ক precompute কের রাখবা যখােন িলখা থাকেব n সাইেজর কয়িট polymino আেছ যারা *w × h* সাইেজ আঁেট আর তারা rotation, flip বা translation সােপে একই িজিনস দয় না।

UVa 225 Golygons

**সমস া:** একিট ি েড িকছু িকছু িব েত যাওয়া িনেষধ। সসব িব ইনপুেট দওয়া থাকেব। এখন তামােক মাট n ধাপ িদেত হেব (*n ≤* 20)। থম ধােপর সাইজ 1, ি তীয় ধােপর সাইজ 2 এরকম কের n ধাপ। তুিম করেব (0*,*0) হেত এবং থম ধাপ তুিম যেকােনা িদেক িদেত পােরা। এই ধাপ দবার পর তামােক অব ই ডােন বা বােম ঘুরেত হেব। যমন তুিম 1 ধাপ িদেল positive y axis বরাবর। এরপর তামােক হয় positive x অথবা negative x axis এর িদেক ঘুের তাকােত হেব এবং সই িদেক তুিম তামার ি তীয় ধাপ দেব (2 সাইেজর)। আবােরা তুিম ডােন বা বােম ঘুরেব এবং সই িদেক 3 সাইেজর ধাপ িদেব। এভােব n বার। খয়াল করেত হেব এই ধাপ দবার সময় যন তুিম িনিষ িব েত বা িনিষ িব র উপর িদেয় না যাও। বলেত হেব এমনভােব n ধাপ দওয়া স ব িকনা যন কােনা িনিষ িব েক শ না কের আবার (0*,*0) ত িফের আসা যায়।

**সমাধান:** থম ধাপ যিদ তুিম উপের বা িনেচর িদেক দাও, তাহেল সকল িবেজাড়তম ধাপ তুিম হয় উপের নাহয় িনেচ িদেব আর জাড়তম ধাপ হয় বােম না হয় ডােন দেব। একই ভােব যিদ থম ধাপ তুিম বােম বা ডােন দাও তাহেল িক হেব তা বুঝেতই পারছ। আপাতত ধের নাও আমরা থম ধাপ উপেরর িদেক িদেয়িছ। এখন মেন কর তুিম িকছু ধাপ দবার পর (*x, y*) এ আছ। এখন উপর

৭৭

িদেক *d* ধাপ দওয়া মােন (*x, y* + *d*) ত যাওয়া, এভােব ডান বাম িনচ সব িকছুর জ আসেল তুিম coordinate এর িহসাব িনকাশ করবা। এখন খয়াল কর, তামার কােছ যত িবেজাড় ধাপ বািক আেছ সই সকল িবেজাড় ধােপর সাইজ িল যাগ কের যিদ দেখা তা *abs*(*y*) এর থেক ছাট, এর মােন যতই চ া কর তুিম *y* = 0 করেত পারেব না। একই ভােব তুিম সকল জাড় ধােপর সাইজ যাগ কের দখেব তার মান *abs*(*x*) এর থেক ছাট হয় িকনা। হেল *x* = 0 করেত পারেব না। এটা একটা optimization. এছাড়াও খয়াল কর backtrack এর সময় যখন তুিম ধাপ দেব তখন জানার েয়াজন হেব য (*x, y*1) হেত (*x, y*2) (বা একই ভােব x axis বরাবর) এর মােঝ কােনা িনিষ িব আেছ িকনা। তুিম চাইেল িত row এবং column এর জ একিট consecutive sum এর অ াের রেখ এই তথ O(1) এ বর কের ফলেত পােরা। এই কয়িট optimization যিদ ব বহার কর তাহেলই আমার মেন হয় এই সম া সমাধান হেয় যােব।

৫.১ অ শীলনী

৫.১.১ সম া

EasyX UVa 140 Bandwidth X Uva 211 The Domino Effect

X UVa 818 Cutting Chains X UVa 690 Pipeline Scheduling X UVa 12113 Overlapping Squares X UVa 1575 Factors X UVa 11214 Guaring the Chessboard X UVa 1533 Moving Pegs X UVa 817 According to Bartjens X UVa 11882 Biggest Number X UVa 12569 Planning mobile robot on Tree (Easy Version)

Medium

X UVa 1354 Mobile Computing X UVa 1603 Square Destroyer X UVa 12107 Digit Puzzle X UVa 11846 Finding Seats Again X UVa 11694 Gokigen Naname X UVa 10384 The Wall Pushers X UVa 10837 A Research Problem X LOJ 1121 15 Puzzle X LOJ 1143 Knights in FEN X LOJ 1397 Sudoku Solver

৫.১.২ িহ ট

**UVa 140:** মা 8 িট নাড। তাই তামরা all possible permutation বর কের চক কর- ত পােরা। এজ backtrack করেত পােরা আবার চাইেল algorithm হডার ফাইেলর next permutation ও ব বহার করেত পােরা।

**UVa 211:** তুিম একিট িনিদ order এ domino বসােব তাহেলই হেব। যমন (1*,*1), (1*,*2), (1*,*3), *...* (1*,c*), (2*,*1) *...* (2*,c*) *...* (*r,*1), *...* (*r, c*) এই অড াের তুিম cell িলেত যােব এবং একিট domino বসােনার চ া করেব। তাহেল িবধা হেলা িতবার তামার হােত মা িট option. হয় এই ঘেরর সােথ তার পােশর ঘেরর, অথবা এই ঘেরর সােথ িনেচর ঘেরর। আর যিদ এই ঘর আেগ থেকই পূরণ হেয় থােক তাহেল তা এটা চ া করার কােনা মােন নই।

**UVa 818:** সম ািট বশ র, িক এর statement টা ambiguous। িক এর পরও িদলাম। কান কান link খুলেত হেব এর উপের bruteforce করার পর একিট graph theory বা adhoc বা greedy (কীভােব একিট সম া এেতা ক াটাগিরর হেত পাের?- আসেল এজ ই আমার এই সম ািট ভাল লেগেছ এবং তামােদর জ এিট িদলাম) সম া সমাধান করেত হয়। এই পিরবিত ত সম ািট বশ র। আমার মেন হয় তামােদর এিট িনেজ থেকই সমাধান করা উিচত।

৭৮

**UVa 690:** খুব একটা কিঠন না। িক কাড একটু ঝােমলার মেন হেত পাের। Reservation table ইনপুট নবার পর থম কাজ হেব এেক কয় ঘর shift কের বসােনা স ব তা বর করা। n ঘর shift কের তা বসােনা স ব কারণ n ঘর সরােল তােদর মােঝ আর কােনা overlap ই থােক না। আর কান কান shift কের বসােনা স ব সিট বর কের রাখেত হেব েতই। এরপর এর উপর িভি কের backtrack. িক এই য বসােনা স ব িকনা বর করা, এই অংশটু আমরা কীভােব খুব ত করেত পাির? আমরা চাইেল bitmask ব বহার করেত পাির। মেন কর আমােদর কােছ একিট bitmask আেছ। এখন আমরা জানেত চাি reservation table ক c ঘর সিরেয় বসােল overlap কের িকনা। এটা আসেল c ঘর shift কের bitwise and কের দখার মত।

**UVa 12113:** র সম া, যিদও কাড একটু কিঠন হেত পাের। থমত খয়াল কর তুিম মা 9 ভােব 2 *×* 2 square ক বসােত পারেব। আর বলাই আেছ য 6 বােরর বিশ বসােব না। অথ াৎ মাট 96 *<* 106 ভােব বসােনা স ব। তরাং তুিম চাইেল েতই সব ভােব বিসেয় বিসেয় shape িল hash কের রাখেত পােরা। অথবা েতই hash এর ঝােমলায় না িগেয় সকল shape বর কের একিট অ ােরেত রেখ িমিলেয় দখেত পােরা ac হয় িকনা।

**UVa 817:** 48 িক খুব একটা বিশ না। অথ াৎ তামরা থম 8 িট character এর পর কী বসাবা (বা হয়েতা বসাবা না) তার উপর bruteforce করেলও হেব।

**UVa 11882:** একিট heuristic হেত পাের তুিম কােনা জায়গায় পৗঁছানর পর দখা য আর কয়িট জায়গায় খুব জাড় যেত পারেব (ধর bfs কের)। IDA\* মেন হয় না করাই ভাল কারণ তামােক সবেচেয় বড় সংখ া বর করেত হেব, এখােন goal িক িনিদ না। অথ াৎ তুিম কােনা একটা সংখ া দেখই বুঝেত পারবা না য এটাই উ র িকনা। তুিম সব সংখ া দখার পর বুঝেব কানটা উ র। এখন মেন কর তুিম ইেতামেধ ই 2345 পেয় গছ। আর এখন 12 নবার পর দখছ আর িট অংেকর বিশ পাওয়া যােব না। এর মােন আমােদর আর এিগেয় লাভ নই। এভােব pruning (pruning মােন হেলা backtrack এর সময় আর না এিগেয় িফের যাওয়া) করেলই হেয় যাবার কথা।

**UVa 1354:** মা 6 িট িজিনস থাকেত পাের। তাই তুিম আসেল সকল ভােব tree বানােনার চ া করেত পােরা। তুিম dynamic programming করার মত কের backtrack করেল িক হেব না। অথ াৎ, যিদ তুিম এরকম কর- root এ তুিম সব িজিনস িদেল, এখােন তুিম িস া িনেল ডােন ক যােব বােম ক যােব, এরপর তুিম বােম আর ডােন recursive call করেল ঐ সব িজিনস িদেয়- না তা হেব না। কারণ বােমর িজিনস িদেয় িবিভ ভােব tree বানােনা যায় আবার ডােনর িজিনস িল িদেয়ও অেনক ভােব tree বানােনা যায়। তরাং তুিম বােমর কান tree এর সােথ ডােনর কান tree ক মলােব সটা একটা কিঠন ব াপার। তেব াঁ, তুিম যিদ বাম বা ডান যেকােনা িদেকর জ যিদ যত ভােব tree বানােনা যায় তােদর িতিটেকই (বা তােদর যই information িল তামার আসেল লাগেব) return করেত পােরা তাহেলই সম া সমাধান হেয় যােব। কােনা নােডর জ আসেল ধু এটু জানেলই হয় য তার subtree ত থাকা নাড িল বাম িদেক এবং ডান িদেক কতদূর পয যায়। তাহেল িদেকর মাট ভর আর িদেকর subtree িট িদেক কতদূর যায় এই িট তথ জানা থাকেল বত মান নােডর জ ও এই তথ তির কের return করেত পারেব।

**UVa 1603:** িতিট ম াচ কািঠর জ িলেখ রাখেব কত িল square এর border এ সই কািঠ আেছ। Backtrack এ কািঠ িলেক তার উপর িদেয় যাওয়া square এর সংখ ার িভি েত বড় হেত ছাট ত সট করেব এবং এই order এ তােদরেক remove করার চ া করেব। কন? াভািবক ভােব িচ া কর, যই ম াচ কািঠ সরােল বিশর ভাগ square ন হেব তােক সরােনাই তা সাধারণত লাভ জনক তাই না? আর িক optimize করা যায়? যিদ দেখা এখন যসব কািঠ বািক আেছ তােদর িতিট িদেয় এেকর বিশ square যায় না তাহেল যকয়িট square বািক আেছ িঠক তত সংখ ক কািঠ তামােক সড়ােত হেব। তরাং আর backtrack না কের তুিম কয়িট square বািক আেছ সই count দখেত পােরা। তেব যেহতু n এর মান কম তাই এসব optimization নাও করা লাগেত পাের। তাও এসব িবিভ রকম optimization জানা থাকা ভাল, কােজ লেগ যেত পাের অ কােনা সম ায়।

৭৯

৮০