Recursion in java factorial

Here we will write programs to find out the causes of a number using recursia. Program 1: The program will ask the user for the input number. After the user provides the input, the program calculates the factors for the input number provided. * @author: BeginnersBook.com * @description: The user will enter the 10 elements * and the program will vaccinate them in the array and * will display their amount. */ java.util.scanner import; FactorialDemo class{ primary public static space (string args[]){ //Scanner object for capturing input scanner user scanner user scanner = new scanner (System.in); system.out.println(enter number:); stored value entered in int num variable = scanner.nextInt(); user is called defined function fact int factorial = fact(num); System.out.println (factorial of number entered is: +factorial); Static fact (int n) { output int; if(n==1){ return 1; } //recursive: a function that calls itself!! output = fact(n-1)* n; return output; } Output: Enter the number: 5 factorial of a number entered is: 120 Plan 2: If you do not want user intervention and simply want to specify the number in the program itself then reference this example. Class FactorialDemo2{ Primary Public Static Space (String Argim[]){ int factorial = Fact(4); System.out.println(Factorial of 4 is: +Factor); Static Fact (int n) { Output int; if(n==1){ return 1; }/Recursive: A function that calls itself!! output = fact(n-1)* n; return output; } Output: Factorial of 4 is: 24 Consider the following recursive function. Public static int mystery (int a, int b) { if (0 == b) return 0; otherwise return + mystery(a, b-1); } Will the function in the previous exercise end for each integer pair A and B between 0 and 100? Give a high-level description of the mystery repetations(a, b), given integers A and b between 0 and 100. Answer: Mystery(1, 7) = 1 + Mystery(1, 6) = 1 + (1 + Mystery(1, 5)) = ... 7 + Mystery(1, 0) = 7. Answer: Yes, the base case is b = 0. Successive recursive calls reduce the B by 1, giving it a direction from the base case. The mystery of the function(a, b) returns * b. Students with a mathematical tendency can prove this fact by induction using the identity ab = a +a(b-1). Consider the following function. What does mystery (0, 8) do? public static space mystery (int a, int b) { if (a!= b) { int m = (a + b) / 2; mystery(a, m); StdOut.println(m); mystery(m, b); } Answer: Infinite Loop. Consider the following function. What does mystery (0, 8) do? public static space mystery (int a, int b) { if (a!= b) { int m = (a + b) / 2; mystery(a, m - 1); StdOut.println(m); Mystery(m + 1, b); } Answer: Stack overflow. Repeat the previous exercise, but replace if (a!= b) if (a. what does mystery(0, 8) do? public static int mystery (int a, int b) { if (== b) StdOut.println(a); other { int m1 = (a + b) / 2; int m2 = (+ b + 1) / 2; mystery(a, m1); b); } What does the following function think? public int f(int n) { if (n == 0) returns 0; if (n == 1) returns 1; if (n == 2) returns 1; Return 2*f(n-2) + f(n-3); write a Fibonacci2 program.java that takes an N command-line argument prints the first N Fibonacci numbers by using the following alternate setting: F(n) = 1 if n = 1 or n = 2 = F(n+) 1)/2++ + F((n-1)/2 if n is odd = F(n/2 + 1)2 - F(n/2 - 1)2 if n is even the largest Fibonacci number that can be calculated in less than a minute using this setting? Compare this to Fibonacci.java. Write a program that takes an N command-line argument and prints the first N Fibonacci numbers using the following method offered by Dijkstra: F(0) = 0 F(1) = 1 F(2n-1) = F(n-1)^1 2 + F(n)^2 F(2n) = (2F(n-1) + F(n)) * F(n) prove by mathematical induction that the alternative settings of the Fibonacci function given in the previous two exercises are equal to the original setting.java Command N and print the first N Pell numbers: p0 = 0, p1 = 1, and n &gt;= 2, pn = 2 pn-1 + pn-2. Print the ratio of consecutive terms and compare to 1 + sqrt(2). Consider the following function from program recursors.java: public static space mystery (int n) { if (n == 0 |n == 1) return; mystery(n-2); (n); Mystery(n-1); } What does mystery(6) print? Hint: First understand what mystery(2), mystery(3), and so on print. What happens in the previous exercise if the base case is replaced by the following statement? Consider the following recursive functions. ריבוע int סטטי ציבורי (int n) { אם (n == 0) להחזיר מרובע ;0להחזיר(n-1) + 2 *n - 1; } קוביית int סטטי ציבורי (int n) { אם (n == 0) החזר ;0 להחזיר קוביה(n-1) + 3*(מרובע(n)) - 3*n + 1; מה ההערכה של פונקציות רקורסיביות הדדיות. שקול את הזוג הבא של פונקציות קוביה(123)? קוביה(5)? קוביה(5)? מהו הערך של ריבוע(5)? g(g(2)? int f(int n) ציבורי { int g(int n) אם } ציבורי (n == 0) אם ;0 מחזיר (n == 0) החזר ;0 מחזיר f(n-1) + g(n-1); החזר g(n-1) + f(n); } של קטנים ערכים עבור) כי לוודא כדי תוכנית כתוב n) את סכום הקוביות של n הראשון מספרי פיבונאצ'י F(0)^3 + F(1)^3 + ... + F(n)^3 שווה (F(3n+4) + (-1)^n * 6 * f(n-1)) / 10 , where F(0) = 1, F(1) = 1, F(2) = 2, and so on. Changes by vibrant and snoury. Given two integers ≤ b, write a sequence plan.java that becomes b by a minimum sequence of incremental addition (insert 1) and action actions (multiply by 2). For example, % java sequence 5 23 23 = (((5* 2 + 1) * 2 + 1) % java sequence 11 113 113 = ((((((11 + 1) + 1) + 1) * 2* 2 * 2 + 1) the hamard matrix. Write a Hadamard recursive program.java that photographs an n command-line argument and outlines an N-by-N Hadamard format in which N = 2n. Do not use an array. The Hadamard format is obtained by aligning 4 copies of the N-by-N format in the form of a 2-by-2 grid, and then reversing the colors of all squares in the lower-right N-by-N copy. The N-by-N Hadamard H(N) matrix is a Boolean matrix with an unusual property that all two rows differ exactly in N/2 bits. This property makes it useful for formatting error correction codes. Here are the first matrices of the Hammard. Eight queens problem. In this exercise, you can solve the classic 8-queen problem: place 8 queens on an 8-by-8 chessboard, so that there are no two queens in the same row, column, or diagonal. There's eight! = 40,320 ways in which no two queens are placed in the same row or column. Any persurances p[] of integers 0 to 7 gives such a position: put the Queen I in line i, column p[i]. Your Queens program.java should take an integer n command-line argument and disperse all solutions to the n-queens problem by drawing the position of the queens in ASCII as the two solutions below. Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Kew, Kew, Kew, Q, Kew, Kew, Kew, Kew, Q Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q* Q* Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q, Q* * Q * * Hint : To determine whether setting q[n] = I conflict with q[0] to q[n-1] if q[i] equals q[n]: Two queens are placed in the same column if q[i] - q[n] equals n - i: two queens are on the same great diagonal if q[n] - q[i] equals n - i: two queens are on the same minor diagonal another 8 kingdom resolves. Queens2 program.java solves the 8 queens problem by implicitly coordinating all n! Change (instead of location n^n). It's based on program .java. The algorithm and the edict is the probability that two numbers selected from a large random set of numbers have no common factors (other than 1) is 6/η2. Use this idea to evaluate Robert Matthews and use the same idea to evaluate β by taking the set of numbers to be a function of the positions of stars in the sky. Hanoi Towers Version II. (Knuth-Grahame and Pathshnik) solve the original Hanoi Towers problem, but with the additional restriction that you are not allowed to transfer directly to disk from A to C. How many action moves to solve a problem on n disks? HINT: Move the smallest n-1 disks from A to C recursively (without any direct A to C movement), move the n disk from A to B, Move the smallest N-1 disks from C to A (without any direct A through C moves), move disk N from B to C, and move the smallest disks from A to C recursively (without any direct A to C movement). Hanoi Towers Version III. Repeat the previous question but slon moves of A to C and C to A. So, each move must include the 4-peg Hanoi Towers. Suppose you have a fourth stake. What is the least number of moves needed to move a stack of 8 disks from the leftmost ply Peg? Answer. Finding the shortest solution in general has remained an open problem for more than a century and is known as a rave puzzle. Another complicated recursive function. Consider the following recursive function. What is f(0)? Public static character (int x) { if (x &gt; 1000) returns x - 4; otherwise the refund f(f(x+5);} checks whether n is a Fibonacci number. Write a function to see if n is a Fibonacci number.java hint: A positive intejarity is a Fibonacci number if and only if one of them (5*n*n + 4) or (5*n*n - 4) is a perfect square.java expr public static string (double p) { r double = Math.random(); if (r complicated recurrence. set F(n) so that F(0) = 0 and F(n) = n - F(n-1).). Solution: The answer is related to the Fibonacci sequence and the representation of a bearddorf of a number. von Neumann. : For i = 0, it's the empty set; For &gt; 0, this is the set containing von Neumann integers 0 to i-1. 0 = {} = {} 1 = {0} = {{}} 2 = {0, 1} = {{}, {{}} 3 = {0, 1, 2} = {{}, {{}, {{}} כתיבת תוכנית Ordinal.java עם של פון נוימן. זוהי שיטה להגדרת תארנים בתיאוריית הסט. תת-רצף של מחרוזת. כתוב תוכנית תת-שורת N -ומחזיר ייצוג מחרוזת של ה N שלוקח מספר שלם לא מייצג ()vonNeumann פונקציה רקורסיבית .java ו מחרוזת של פקודה שורת ארגומנט שלוקחת - k של ההתבודדות תת כל את ומדפיסה שלם פקודה שורת של ארגומנט s באורך k. % java Subsequence abcd 3 abc abd acd bcd Interleaving שתי מחרוזות. בהינתן שתי מחרוזות s ו- t להדפיס את כל של תווים שונים, כאשר (M+N)! לא,לא,לא,לא,לא,לא שבין, אני לא יודע אם אני יכול לעשות את זה. לדוגמה, אם N ו- M הם מספר התווים בשתי המחרוזות s = ab t = CD abCD CabD aCbD CaDb aCDb CDab CDab בינארי GCD. כתוב תוכנית בינאריGCD.java שמוצאת את הממחז המשותף הגדול ביותר של שני מספרים שלמים חיוביים באמצעות אלגוריתם GCD בינארי : gcd(p, q) = p if q = 0 q if p = 0 2 * gcd(p/2, q/2) if p and q are even gcd(p/2, q) If p is even and q is odd gcd(p, q/2) if p is odd and q is even gcd((p-q)/2, q) if p and q are odd and p&gt;= q gcd(p, (q-p)/2) if p and q are odd and p &lt; q partitions Write a .java the N-positive integer as a command-line argument and print all partitions of N. N partition is a way to write N as a trache of positive integers. Two sums are considered the same if they differ only in the order of their constituent summaries. Partitions arise in symmetrical polynums and in the theory of group representation in mathematics and physics. % Java Partition 4 % Java Partition 6 4 6 3 1 5 1 2 2 2 2 1 1 4 1 1 1 1 1 1 3 3 3 3 3 21 3 11 2 2 2 2 1 2 11 11 11 11 11 11 11 11 11 Johnson-Trotter perverts. Write a Plan Johnson Trotter.java so Command-line argument in whole number n and print all n! Perverts of an inte a number 0 through n-1 in such a way that subsequent perverts differ only in one adjacent location (similar to how the gray code passes on joins in such a way that subsequent combinations differ in just one second). % java JohnsonTrotter 3 012 (2 1) 021 (1 0) 201 (2 1) 210 (0 1) 120 (1 2) 102 (0 1) lexicographical order permanes. Write a change planLex.java that has a command-line argument N and prints the entire N! Perswing an inte intewer number 0 to N-1 in lexicographic order. % java permanesLex 3 012 021 102 120 201 210 crazy. Crazy is a p[] of integers from 0 to N-1 so that p[i] is not equal to i for each i. For example, there are 9 vibrant when N = 4: 1032, 1230, 1302, 2031, 2301, 2310, 3012, 3201, 3210. Write a program to count the number of N-sized disturbed by using the following recurrence: d[N] = (N-1) (d[N-1] + d[N-2]), where d[1] = 0, d[2] = 1. The first conditions are 0, 1, 2, 9, 44, 265, 1854, 14833, 133496, and 1334961. Tribunachi numbers. Tribonacci numbers are similar to Fibonacci numbers, but each term is the sum of the previous three terms in a row. The first conditions are 0, 0, 1, 1, 2, 4, 7, 13, 24, 44, 81. Write a plan for calculating tribuneci numbers. What are the follow-up terms of the ratio? Answer. Root x^3 - x^2 - x - 1, which is approximately 1.83929. Sum of the first Fibonacci numbers. It has been shown by induction that the sum of the first n n fibonacci numbers F(1) + F(2) + ... + F(N) is F(N+2) - 1. Combined gray code. Print the entire combination of k of n items in such a way that consecutive combinations differ exactly in one element, for example, if k = 3 and n = 5, 123, 134, 234, 124, 145, 245, 345, 135, 235, 125. Hint: Use the gray code, but print only the integers with exactly 1 k in their binary representation. The labyrinthine generation. Create a maze using divide and conquer: start with a rectangular area without walls. Select a random grid point in the rectangle and build two perpendicular walls, dividing the square into 4 sub-neighborhoods. Select 3 of the four areas at random and open one cell hole at a random point in each of 3. Repeats until each sub-purpose has a width or height of 1. Plasma clouds. PlasmaCloud program.java takes an N command-line argument and produces a random N-by-N plasma fractel using the midpoint displacement method. Here's an example of 800 over 800. Below is a reference, including a simple 1D version. NOTE: Some visualizations stand out in parallel with the x and y varieties. Write a recursive plan to draw a shed or a tree, as in this fractel-lace demo. Intification number definition partition. Use Memos to develop a program that resolves the partition problem that is set for positive Values. You can use an array that is the sum of input values. Voting power. John F. Benzaff III proposed a rating system for each coalition in a block voting system. Suppose the party I control votes. An absolute majority of votes are required to accept or reject an offer. Party I's voting power is the number of minority coalitions it can join and turn it into a winning majority coalition. Write a VotingPower program.java that takes a list of coalition weights as a command-line argument and prints the voting power of any coalition. HINT: Use the clipboard .java starting point. Timing in two parallel machines. This schedule .java the program takes a command-line N argument, reads in a real N number of standard input, and partitions them into two groups so that the difference between them is minimized. A $10,000 conway streak. Consider the following recursive function. f(n) = f(f(n-1)) + f(n-f(n-1)) for n &gt; 2 and f(1) = f(2) = 1. Computer f(3). Write a Java program to calculate the first 50 values of f(n) sequentially Hofstadterâ€Conway $10,000. Use dynamic programming. This sequence has many fascinating qualities and connects to Pascal's triangle, the Distribution of The Goos, Fibonacci numbers and Catalan numbers. Repeat running time. Use dynamic programming to calculate a T(N) value table, where T(N) is the solution to the next separation and differentiation. T(1) = 0, T(N) = N + T(N/2) + T(N - N/2) if N &gt; 1. Optimizing a gas station. You drive from Princeton to San Francisco in a car that gets 25 miles a gallon and has a fuel tank capacity of 15 gallons. Along the way, there are N gas stations where you can stop for fuel. Gas station I d[i] miles into the ride and sells gas for p[i] dollars per gallon. If you stop at the station I'll for gas, you must completely fill your tank. Let's say you start with a full container and that d[i] are integers. Use dynamic programming to find a minimum cost sequence of stops. Thein Deif. The Unix DIFF program compares two files line by line and prints places where they differ. Write a Diff program.java which reads two files specified on the command line one line at a time, calculates the LCS in the row sequence that make up each file and prints rows that correspond to discrepancies in the LCS. The longest common subsedior of 3 strings. Given 3 strings, find the longest shared sub-way through dynamic programming. What is the runtime and memory usage of your algorithm? Making a difference. Given the $100 bills, $50 bills, $20 bills, $10 bills, $5 bills, $50 bills, $20 bills, determine whether a change can be made to Ann St. Hint: Backpack problem. (Greedy also works.) Making a difference. Suppose you're a weird cashier. When currencies are found: 1, 3, 8, 16, 22, 57, 103, and 526 cents (or more usually d0, d1, ..., dN-1. Describe a dynamic programming algorithm to make a change for c cents using the slightest number of coins. hint: The greedy algorithm won't work since the best way to change 114 cents is 57 + 57 instead of 103 + 8 + 3. The longest growing sequence. Find the longest and most rigorously growing sub-way. Hint. Calculate the longest shared sublist between the original array and a sorted version of the array in which duplicate copies of an insam number are removed. Longest common line-up sequence. Computational biology. Given two sequences of 64-bit N integers, find the longest growing sub-sequence common to the two sequences. Select a task with spaces. Work I have s_i start time f_i, time p_i time and space. Diff. Write a program that reads two files and prints their dependings. Treat each row as an icon and LCS account. Print these lines in any file that are not in the LCS. Backpack problem. Pod.java P.O.B. Write a program that takes a command-line N argument, reads text from the regular input, and prints the text, nicely formatted with a maximum of N characters per line. Use dynamic programming. Viterbi algorithm. Given a deliberate graph in which each end is marked with an icon from a finite alphabet. Is a path value from one unique codex matching characters in the s string? Dynamic programming. A(i, v) = 0 or 1 if there is a path from x to v that consumes the first characters of s. A(i, v) = Maximum (A(i-1, u) : (u, v) in E labeled with [i]). Viterbi algorithm. Speech recognition, handwriting analysis, computational biology, hidden models of Markov. Assume that each edge that leaves v has a probability of passing p(v, w). A probability of a path is a product of the probability in that path. What's the most likely way? Dynamic programming. Smith-Waterman algorithm. Aligns a local sequence. Binomial coefficient (brutal force). The binary coefficient C(n, k) is the number of ways to select a subset of k-elements from a group of n elements. It's going up in probability and statistics. One formula for binomial coefficient computing is C(n, k) = n! I don't know if I can do this. (n-k)!) This formula is not so suitable for direct calculation because the interim results may be overflowing, even if the final answer is not. For example C(100, 15) = 253338471349988640 fits 64-bit long, but the binary representation is 100! It's 525 bits long. Pascal's identity expresses C(n, k) in terms of a smaller binomial coefficient: SlowBinomial.java fails spectacularly for medium n or k, not because of overflow, but because the same sub-issues are resolved repeatedly. Do not run this code for large public static and long binomial inputs (int n, int k) אם (k == 0) להחזיר 1; אם (n == 0) החזר בינומי 0; מחזיר בינומי(n-1, k) + בינומי(n-1, k-1); } כתוב תוכנית בינומית (תכנות דינמי). מקדם בינומי.java של שורת הפקודה הנושלת שני ארגומנטים n ו- k ומשתמשת בתכנות דינאמי מלמטה למעלה כדי לחשב C(n, k). מחלקה ציבורית Comb2 עמוד 2 { חלל סטטי פרטי להראות שילוב (int[] s) { עבור (int i = 0; i &lt; s.length; i++) StdOut.print(s[i] + חלל סטטי ציבורי ליצור { לאחר ש ,2015-ב (int[] s, int מיקום, int nextInt, int k, int n) { אם (מיקום == k) { showCombination(s); עבור { ;החזרה (int i = nextInt; i &lt; n; i++) { s[position] = i; generate(s, position + 1, i + 1 , k, n); } חלל סטטי ציבורי ראשי (String[] args) { int n = מספר שלם.parseInt(args[0]); int k = מספר שלם.parseInt(args[1]); int[] s = int[k]; generate(s, 0, 0, k, n); } } }