

УДК 519.1

DOI: 10.21779/2542-0321-2023-38-2-39–47

А.М. Магомедов¹, С.А. Лоренс², О.И. Челябинна²

Алгоритмический подход к задаче перечисления разбиений прямоугольной полосы

¹ Дагестанский государственный университет; Россия, 367000, г. Махачкала, ул. М. Гаджиева, 43а; tagomedtagirl1@yandex.ru;

² Российский государственный университет туризма и сервиса, Институт сервисных технологий; Россия, 141221, Московская обл., Пушкинский р-н, д/н Черкизово, ул. Главная, 99; lawrencenko@hotmail.com, olga-chelyapina@mail.ru

Аннотация. Ранее авторами был описан двухшаговый метод подсчета количества разбиений прямоугольной полосы с целочисленными длинами сторон на плитки размером 1×2 . В данной статье изложен алгоритм его реализации, разработано программное обеспечение.

Построено продолжение метода первой светлой клетки, которое упрощает вычисление коэффициентов прямой рекурсии. Приведены результаты вычислительных экспериментов.

Ключевые слова: рекурсия, формула, алгоритм, программа, перечисление.

Введение¹

Формулировка задачи

Задача. Найти количество способов $T(m, n)$ разбиения прямоугольника $M(m \times n)$ из m столбцов и n строк на прямоугольники 1×2 (плитки), каждую из которых можно уложить горизонтально или вертикально.

Задача подсчета $T(m, n)$ эквивалентна задаче нахождения количества совершенных паросочетаний в решетчатом графе. Задача нахождения количества совершенных паросочетаний в случае графов общего вида принадлежит классу #P-полных задач, даже если ограничиться двудольными графами. В 1961 г. Кастелейну [1], Темперли и Фишеру [2] удалось построить эффективный алгоритм для решения задачи в случае решетчатых графов. Интересно, что понятие полиномиального алгоритма возникло несколько позже. Задача возникает в прикладных вопросах физики и химии [3–4]. Так, некоторые классы химических соединений успешно синтезируются только тогда, когда графы соединений в топологической модели молекулы имеют совершенное паросочетание, а компоненты некоторых семейств соединений тем стабильнее, чем больше совершенных паросочетаний содержатся в их графах.

В ряде публикаций отмечается, что решение было получено авторами [1–2] одновременно и независимо (причем одним и тем же методом!) в процессе исследования вопросов термодинамики потоков жидкости.

Операции с плавающей запятой затрудняют точные вычисления по следующей формуле Кастелейна, полученной в [1]:

$$T(m, n) = 2^{mn/2} \prod_{j=1}^m \prod_{k=1}^n \left(\cos^2 \frac{\pi j}{m+1} + \cos^2 \frac{\pi k}{n+1} \right)^{\frac{1}{4}}. \quad (1)$$

¹ Работа выполнена при поддержке Отдела математики и информатики ДФИЦ РАН.

Различным аспектам задачи посвящены десятки статей (см., например, [1–16]). Добавим, что в [11, гл. 7] задача рассмотрена в качестве приложения производящих функций. Для небольших m и n задача была предложена на VI Всероссийской олимпиаде школьников по информатике [11–14].

Если m и n – нечетные, искомое разбиение не существует, поэтому будем полагать, что один из этих параметров (например, m) имеет четное значение. Легко показать, что при $m = 2$ решение задачи дается числами Фибоначчи: $T(2,0) = T(2,1) = 1, T(2,n) = T(2,n-1) + T(2,n-2)$ при $n \geq 2$ (мы не будем останавливаться на разъяснениях утверждений типа $T(2,0) = 1$).

В литературе уделяется внимание вариациям задачи, когда из M вырезано несколько граничных клеток. В этом случае исходная фигура называется (прямоугольной) *полосой*, мы сохраним термин также и для случая, когда число удаленных клеток равно нулю. Например, фигуру, полученную удалением из шахматной доски двух клеток из противоположных по диагонали углов, нельзя разрезать на плитки, каждая из которых представляет собой две соседние (по стороне) разноцветные клетки доски; для такого заключения достаточно обратить внимание на тот факт, что в оставшейся фигуре черных и белых клеток не поровну. Фигуру, полученную удалением любых двух разноцветных граничных клеток шахматной доски, можно разрезать на плитки.

Два продолжения метода первой светлой клетки

Эквивалентные вершины

Будем полагать, что клетки M пронумерованы слева направо, сверху вниз: 0, 1, ... Следуя терминологии работы [15], множество незанятых пока клеток будем называть *светлой фигурой*, незанятую клетку с наименьшим номером – *первой светлой клеткой* (п. с. к.). Будем укладывать плитки так, чтобы каждая очередная плитка покрывала п. с. к. (метод п. с. к.). Когда значение m фиксировано, будем обозначать число способов разбиения:

- а) прямоугольной полосы M ;
- б) фигуры, полученной из M удалением клеток 0 и 1;
- в) фигуры, полученной из M удалением клеток 0 и 3,

через a_n , b_n и c_n соответственно.

В [15] построено двоичное дерево с корнем в вершине a_n ; потомки вершины v – частичного разбиения (ч. р.) генерируются в результате горизонтальной и/или вертикальной укладки плитки в соответствии с методом п. с. к. Каждая вершина v характеризуется высотой светлой фигуры и границей – ломаной, отделяющей светлую фигуру от остальной части полосы. Высоту h светлой фигуры ч. р. v будем называть *индексом* ч. р., список $(d_1, \dots, d_m)$, где d_k – разность между h и высотой k -того столбца светлой фигуры – *дескриптором* v . Две вершины будем называть *эквивалентными*, если их дескрипторы совпадают с точностью до симметрии (равенство высот не требуется). Из очевидного равенства количества полных разбиений (к. п. р.) родительской вершины сумме к. п. р. ее непосредственных потомков следует простое *правило потомков*: к. п. р. корневой вершины любого поддерева двоичного дерева равно сумме к. п. р. всех листьев этого поддерева.

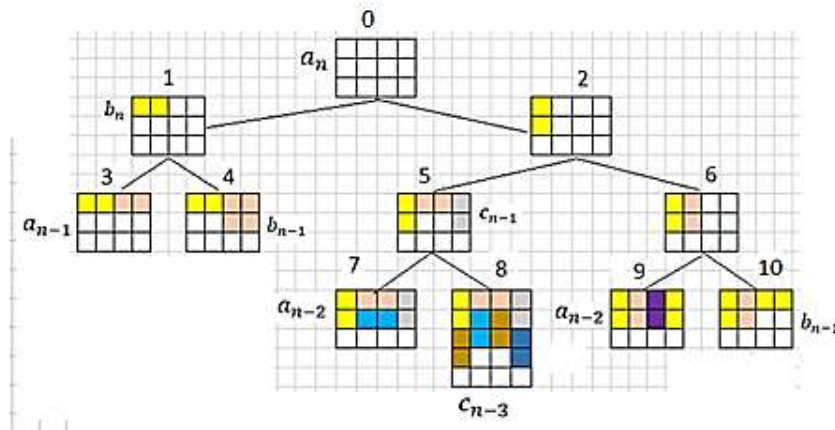


Рис. 1. Завершенное двоичное дерево

Построение двоичного дерева завершается, когда для каждой вновь созданной вершины-листа в построенной части найдется эквивалентная вершина. На рис. 1 эквивалентны: а) вершины 1, 4, 10; б) вершины 0, 3, 7, 9; в) вершины 5, 8.

Первое продолжение метода п.с.к.

Применяя правило потомков к поддеревьям двоичного дерева на рис. 1 с вершинами а) 1, 3, 4; б) 5, 7, 8; в) 0, 1, 2, 5, 6, 9, 10, получим систему взаимно рекуррентных формул:

$$\begin{cases} a_{n-1} = b_n - b_{n-1}, \\ c_{n-1} = a_{n-2} + c_{n-3}, \\ a_n = a_{n-2} + b_n + b_{n-1} + c_{n-1}. \end{cases} \quad (2)$$

Начальные значения a_n, b_n, c_n при $n = 1$ и $n = 2$ легко вычислить:

$$a_1 = 1, a_2 = 5, b_1 = 1, b_2 = 2, c_1 = 1, c_2 = 1.$$

В [15] показано, как из (2) вывести формулу *прямой рекурсии*:

$$a_n = a_{n-1} + 5a_{n-2} + a_{n-3} - a_{n-4}. \quad (3)$$

Прямым линейным однородным рекуррентным соотношением (далее кратко – *прямой рекурсией*) порядка R для последовательности $a_1, a_2, \dots$ будем называть представление произвольного элемента a_r ($r > R$) в виде линейной комбинации с действительными постоянными целочисленными коэффициентами R элементов $a_{r-1}, a_{r-2}, \dots, a_{r-R}$, непосредственно предшествующих a_r :

$$a_r = p_1 a_{r-R} + p_2 a_{r-R+1} + \dots + p_R a_{r-1}, p_1 \neq 0.$$

Второе продолжение метода п. с. к.

Несколько неожиданно, что первая из формул системы (2)

$$b_n = a_{n-1} + b_{n-1}$$

не только позволяет выразить a_n через элементы последовательности $b_1, b_2, \dots$ но допускает и обратное – выразить b_n через элементы последовательности a_1, a_2 . В самом деле, сложив почленно равенства, выписанные для убывающих индексов

$$\begin{aligned} b_n &= a_{n-1} + b_{n-1}, \\ b_{n-1} &= a_{n-2} + b_{n-2}, \\ b_{n-2} &= a_{n-3} + b_{n-3}, \\ &\dots \dots \dots \end{aligned}$$

получим $b_n = a_{n-1} + a_{n-2} + a_{n-3} + \dots$

Это наблюдение лежит в основе следующего способа получения прямой рекурсии. Вместо прерывания построения двоичного дерева на рис. 1 сразу после того, как для каждого его листа обнаружена эквивалентная вершина в построенной части дерева, продолжим его построение и определим, что правило потомков непосредственно формирует прямую рекурсию для a_n без решения системы взаимно рекуррентных формул. В самом деле, из рис. 2:

$$\begin{aligned} a_n = & (a_{n-1} + a_{n-2} + a_{n-3} + \dots) + \\ & + (a_{n-2} + a_{n-4} + a_{n-6} + \dots) + \\ & + (a_{n-2} + a_{n-2} + a_{n-3} + a_{n-4} + \dots). \end{aligned} \quad (4)$$

Прямая рекурсия получена. С целью избавиться от многоточий в записи формулы вычтем из

$$a_n = a_{n-1} + 4a_{n-2} + 2a_{n-3} + 3a_{n-4} + 2a_{n-5} + 3a_{n-6} + \dots$$

равенство

$$a_{n-2} = a_{n-3} + 4a_{n-4} + 2a_{n-5} + 3a_{n-6} + 2a_{n-7} + 3a_{n-8} + \dots$$

Получим: $a_n - a_{n-2} = a_{n-1} + 4a_{n-2} + a_{n-3} - a_{n-4}$ или $a_n = a_{n-1} + 5a_{n-2} + a_{n-3} - a_{n-4}$.

Задача об оптимальной прямой рекурсии

Если утверждение из работы [16] «...те же коэффициенты, что и в (3), подходят и для прямых рекурсий, описывающих последовательности $b_1, b_2, \dots$ и $c_1, c_2, \dots$ » еще можно было предугадать, то обнаруженная там же неоднозначность прямой рекурсии стала полной неожиданностью.

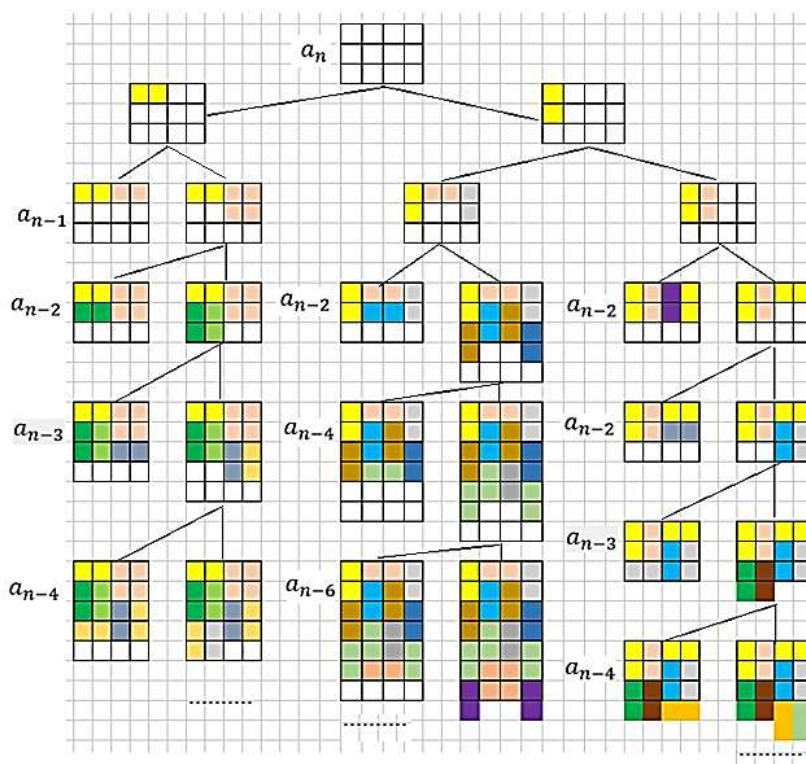


Рис. 2. Продолжение построения двоичного дерева

Задача. Найти для последовательности $a_1, a_2, \dots$ прямую рекурсию, оптимальную по вычислительной сложности (хотя бы для случая $m = 4$).

В формуле $a_n = 6(a_{n-2} + a_{n-3}) - a_{n-5}$ одна операция умножения и две операции алгебраического сложения целых чисел, тогда как в (3) одна операция умножения и три операции сложения.

Алгоритмическое (программное) обеспечение двухшагового метода

Схема вычисления

В [16] рассмотрена задача нахождения при некотором заданном m натурального R (порядок рекурсии) и целочисленных коэффициентов $x_1, x_2, \dots, x_R$, таких, что

$$a_r = x_1 a_{r-1} + x_2 a_{r-2} + x_3 a_{r-3} + \dots + x_R a_{r-R} \quad (5)$$

для каждого $r = R + 1, R + 2, \dots$

R можно выбрать равным $2^{\lceil m/2 \rceil}$. Суть предложенного в [16] двухшагового подхода к вычислению a_r такова. **Первый шаг** – с привлечением любых средств вычислить значения

$$a_1, a_2, a_3, \dots, a_{2R}$$

и для формирования рекуррентной формулы (5) решить с. л. а. у.:

$$\begin{pmatrix} a_R & \dots & a_1 \\ & \dots & \\ a_{2R-1} & \dots & a_R \end{pmatrix} \begin{pmatrix} x_1 \\ \dots \\ x_R \end{pmatrix} = \begin{pmatrix} a_{R+1} \\ \dots \\ a_{2R} \end{pmatrix}.$$

Второй шаг – вычислить числа a_r для произвольного $r = 2R + 1, 2R + 2, \dots$ по полученной рекуррентной формуле, используя только операции сложения и умножения.

Мы предлагаем лаконичный (около одной страницы, включая комментарии) код для реализации двухшагового метода. Для применения достаточно вырезать код в окно системы компьютерной математики Wolfram Mathematica и после команды выполнения (Shift+Enter) в диалоговом режиме ввести значения m и R .

Код программы

```
m = Input[m, "Введите m:"];
maxindex = Input[maxindex, "Введите maxindex:"];
Print[" m = ", m, ", количество требуемых димерных чисел = ", maxindex];
Array[a, maxindex];
key = If[Mod[m, 4] == 0, 1, -1];
R = 2^(m/2); r = R/2;
(*Начальный этап: вычисление 2R+1 начальных элементов последовательности произвольным способом (например, по формуле Кастелейна).*)
Do[{a[k] = Round[N[Product[4*Cos[Pi*i/(k+1)]^2+4*Cos[Pi*j/(m+1)]^2, {i, 1, k/2}, {j, 1, m/2}], 5000]]];
  filename = StringJoin["C:\\1\\", ToString[m], ".txt"];
  PutAppend[StringJoin["a[", ToString[k], "] = ", ToString[a[k]]], filename];
  Print["a[", k, "] = ", a[k]], {k, 1, 2*R+1}];
Print["a[", 2 R+1, "] – контрольное значение (будет перевычислено)"];
(*Основной этап: вычисление элементов с номерами 2R+1, 2R+2, ..., maxindex высокоскоростным алгоритмом с использованием только операции умножения и сложения многоразрядных целых чисел. *)
M = Table[If[j<r, a[R+i-j]+If[key == -1, (-1)^j, 1]*a[i+j], a[i+r]], {i, 1, r}, {j, 1, r}];
(*1 случай: j = r;
в i-той строке с. л. а. у. коэффициентом при x[j] выступает a[i+r];
2 случай: j<r;
```

```

2.1 если m кратно 4, то в i-той строке с. л. а. у. коэффициент при x[j] равен
a[R+i-j]+a[i+j];
2.2 если m четно, но не кратно 4, то в i-той строке с. л. а. у. коэффициент при x[j] равен
a[R+i-j]+(-1)^j*a[i+j];*)
B = Table[a[i]+a[R+i], {i,1,r}];
(* Сформировали матрицу коэффициентов с. л. а. у. *)
Print["----- m = ",m," ----коэффициенты прямой рекурсии:"];
x = LinearSolve[M,B]
PutAppend[x,filename];
PutAppend["The coefficients of direct recursion are calculated!", filename];
(*Решение с. л. а. у. вывели на экран и в файл*)
Do[a[R+i] = Sum[x[[j]]*(If[j<r, a[R+i-j]+If[key == -1, (-1)^j, 1]*a[i+j], a[i+r]]), {j,1,r}]-a[i];
(*Элементы с номерами 2R+1, ..., maxindex вычислили с применением операций * и +*)
PutAppend[StringJoin["a[",ToString[R+i], "] = ", ToString[a[R+i]]], filename];
(*и вывели в файл*)
Print["a[",R+i,"] = ", a[R+i]], {i,R+1, maxindex-R}]
(*и на экран*)

```

Скриншот вывода результатов

```

m=6, количество требуемых димерных чисел=18
a[1]=1
a[2]=13
a[3]=41
a[4]=281
a[5]=1183
a[6]=6728
a[7]=31529
a[8]=167089
a[9]=817991
a[10]=4213133
a[11]=21001799
a[12]=106912793
a[13]=536948224
a[14]=2720246633
a[15]=13704300553
a[16]=69289288909
a[17]=349519610713
a[17] - контрольное значение (будет перевычислено)
----- m=6 ----коэффициенты прямой рекурсии:
Out[12]= {1, 20, 10, -38}
a[17]=349519610713
a[18]=1765722581057

```

О результатах вычислительного эксперимента

Задача о частоте цифр

Вычисления, выполненные для косвенной проверки некоторых утверждений, приведенных в данной статье, нередко проводились с участием сверхбольших чисел. Ограничимся одним примером, причем выпишем лишь начало и конец числа, т. к. для полной записи числа понадобилось бы полтора десятка страниц. При $m = 8$, $n = 50000$,

$$T(m, n) = 5024...66546544059025906 \text{ (число цифр} = 47\,729\text{)}.$$

Обозначим количество вхождений в десятичную запись $T(m, n)$ цифры, которая встречается реже (соответственно чаще) остальных цифр, через $\min(T(m, n))$ (соответственно $\max(T(m, n))$),

$$d(m, n) = \frac{\min(T(m, n))}{\max(T(m, n))}.$$

Наблюдается устойчивая тенденция к росту значений $d(m, n)$ при увеличении n и m , "эффект выравнивания" цифр в десятичной записи $T(m, n)$ при увеличении m и n .

Численные результаты

Чтобы добиться большей отчетливости картины, вычисленные значения $d(m, n)$ были усреднены по "блокам". Каждый блок вычислений соответствует фиксированному значению m ($m = 2, 4, 6, 8, 10$) и содержит по 10 значений $d(m, n)$, вычисленных для возрастающих значений n .

Как видно из таблицы 1, последовательность усредненных значений для $m = 2, 4, \dots, 10$, $n = 1000, 2000, \dots, 10000$ и $10000, 20000, \dots, 100000$ стремится к 1 с увеличением параметров m и n : 0,77, 0,81, 0,82, 0,87, 0,86, ..., 0,92, 0,92, 0,94, 0,94, 0,96 ...

Таблица 1. Выравнивание частот цифр в десятичной записи $T(m, n)$

n	$m = 2$	$m = 4$	$m = 6$	$m = 8$	$m = 10$
1000	0,57	0,56	0,60	0,79	0,71
2000	0,61	0,75	0,72	0,80	0,85
3000	0,74	0,88	0,79	0,86	0,84
4000	0,74	0,77	0,82	0,91	0,86
5000	0,82	0,84	0,83	0,89	0,88
6000	0,77	0,79	0,85	0,90	0,85
7000	0,89	0,88	0,85	0,89	0,91
8000	0,88	0,87	0,88	0,88	0,88
9000	0,86	0,92	0,92	0,90	0,90
10000	0,84	0,84	0,89	0,90	0,92
Усредненные значения	0,77	0,81	0,82	0,87	0,86

n	$m = 2$	$m = 4$	$m = 6$	$m = 8$	$m = 10$
10000	0,84	0,84	0,89	0,90	0,92
20000	0,89	0,91	0,92	0,90	0,95
30000	0,91	0,92	0,92	0,94	0,95
40000	0,92	0,88	0,97	0,96	0,96
50000	0,90	0,93	0,91	0,94	0,96
60000	0,95	0,96	0,97	0,95	0,97

70000	0,91	0,93	0,96	0,93	0,96
80000	0,95	0,97	0,95	0,97	0,97
90000	0,92	0,96	0,97	0,98	0,97
100000	0,96	0,95	0,96	0,96	0,97
Усредненные значения	0,92	0,92	0,94	0,94	0,96

Заключение

В статье найдено эффективное продолжение метода первой светлой клетки, позволяющее вычислить коэффициенты прямой рекурсии (при $m = 4$); сформулированная задача об оптимальной рекуррентной формуле может стать началом нового перспективного направления исследований.

Разработан лаконичный эффективный код для сопровождения предложенного в [16] двухшагового метода. В результате вычислительного эксперимента подтверждается тезис о выравнивании частот цифр в десятичной записи чисел $T(m, n)$ при росте значений m и n .

Литература

1. Kasteleyn P.W. The statistic of dimers on a lattice I: The number of dimer arrangements on quadratic lattice // Physica. 1961. Vol. 27. – Pp. 1209–1225.
2. Temperley H.N.V. and Fisher M.E. Dimer problem in statistical mechanics – an exact result // Phil. Mag. 1961. Vol. 6. – Pp. 1061–1063.
3. Ловас Л., Пламмер М. Прикладные задачи теории графов. Теория паросочетаний в математике, физике, химии. – М.: Мир, 1998. – 653 с.
4. Эллиот В. Монтролл. Статистика решеток // Прикладная комбинаторная математика. Сборник статей / под ред. Э. Беккенбаха. – М.: Мир, 1968.
5. Fowler R.H., Rushbrooke G.S. Statistical Theory of Perfect Solutions, Trans. Faraday Soc. 1937. Vol. 33 – Pp. 1272–1294.
6. Магомедов А.М., Магомедов Т.А., Лавренченко С.А. Взаимно-рекуррентные формулы для перечисления разбиений прямоугольника // Прикладная дискретная математика. 2019. № 46. – С. 108–121.
7. Караваев А.М., Перепечко С.Н. Задача о димерах на цилиндрах: рекуррентные соотношения и производящие функции // Математическое моделирование. 2014. Т. 26, № 11. – С. 18–22.
8. Valiant L.G. The complexity of computing the permanent // Theoretical Computer Science. 1979. Vol. 8. – Pp. 189–201.
9. Klarner D. and Pollack J. Domino tilings of rectangles with fixed width // Discrete Mathematics. 1980. Vol. 32. – Pp. 45–52.
10. Read Ronald C. A note on tiling rectangles with dominoes // Fib. Q. 1980. Vol. 18, no. 1. – Pp. 24–27.
11. Грэхем Р., Кнут Д., Паташник О. Конкретная математика. Основание информатики: пер. с англ. – М.: Мир, 1998. – 703 с.
12. Кирюхин В.М. VI Всероссийская олимпиада школьников по информатике // Информатика и образование. 1994. № 3. – С. 47–50.
13. Волченков С.Г. Задача «Паркет» // Информатика и образование. 1994. № 3. – С. 52–54.

14. Василевский Б. Динамическое программирование по профилю // Московские учебно-тренировочные сборы по информатике. Весна–2006 / под ред. В.М. Гуровица. – М.: МЦНМО, 2007. – 194 с.

15. Магомедов А.М., Лавренченко С.А. Некоторые свойства прямых рекуррентных соотношений для последовательностей димерных чисел // Вестник Дагестанского государственного университета. Сер. 1: Естественные науки. 2022. Т. 37, вып. 1. – С. 51–62.

16. Магомедов А.М., Лавренченко С.А. Вычислительные средства C# для решения задачи перечисления разбиений прямоугольника // Вестник Дагестанского государственного университета. Сер. 1: Естественные науки. 2020. Т. 35, вып. 4. – С. 13–26.

Поступила в редакцию 25 мая 2023 г.

UDC 519.1

DOI: 10.21779/2542-0321-2023-38-2-39–47

Algorithmic Approach to the Problem of Enumerating Partitions of a Rectangular Strip

A.M. Magomedov¹, S.A. Lawrence², O.I. Chelyapina²

¹ Dagestan State University; Russia, 367000, Makhachkala, M. Gadzhiev st., 43a; *magomedtagir1@yandex.ru*;

² Russian State University of Tourism and Service, Institute of Service Technologies; Russia, Moscow region, 141221, Pushkino district, Cherkizovo, Glavnaya st., 99; *lawrencenko@hotmail.com, olga-chelyapina@mail.ru*

Abstract. Previously, the authors described their two-step method of calculating the number of fragmentations of a rectangular bar with 1×2 size. The implementation algorithm is described in this article, and software is developed.

An extension of the first light cell method is proposed, simplifying the calculation of direct recursion coefficients. The results of the computational experiments are presented.

Keywords: recursion, formula, algorithm, program, enumeration.

Received 25 May 2023