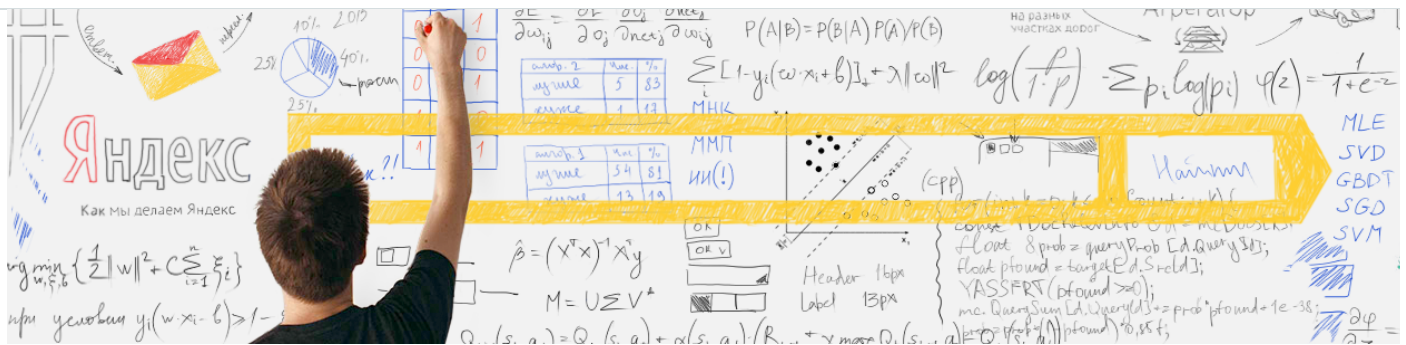




Яндекс 795,02
Как мы делаем Яндекс

4 августа 2014 в 13:51

Разбор финальных задач Яндекс.Алгоритма 2014

Алгоритмы*, Блог компании Яндекс, Спортивное программирование*

1 августа в офисе Яндекса, открывшемся недавно в Берлине, состоялся финал нашего чемпионата по программированию. И его победителем стал известный всем, кто интересуется спортивным программированием, [Геннадий Короткевич](#).

Задания для Алгоритма готовила международная команда. В нее вошли программисты из России, Беларуси, Польши и США. Это специалисты имени М.В. Ломоносова, Университета Карнеги-Меллон, сотрудники Яндекса и Google. В Яндексе задачи составляли разработчики минского киевского офиса, а потом проверяли их на своих коллегах. Один из составителей в прошлом году сам был финалистом Алгоритма. Специаль Хабрахабра мы разобрали с авторами все задачи. Кстати, несмотря на то, что соревнование завершено, вы можете попробовать себя в [вирул контексте](#).



На победу претендовали многие финалисты. Среди них были победители и призеры ACM ICPC и TopCoder Open, разработчики Google и Facebook. В финальном раунде сражались призёры Алгоритма-2013 [Евгений Капун](#) и Ши Бисюнь, чемпион ACM ICPC [Михаил Кевер](#), а также один из самых титулованных спортивных программистов мира [Пётр Митричев](#). В этом году побороться за приз решил также [Макото Созэзимо](#) — составитель заданий для Алгоритма-2013 и администратор TopCoder Open.

Борьба за первое место разгорелась между Геннадием Короткевичем и [Хосакэй Кадзухиро](#) из Токийского университета. Лучший результат — четыре задачи при 66 минутах штрафного времени — показал Короткевич, подтвердив титул чемпиона. Кадзухиро решил столько же задач, набрал больше штрафного времени (90 минут) и занял второе место. Третье место завоевал [Ван Циньши](#) из университета Цинхуа: он решил задачи при 125 минутах штрафа.

Задача А. 1024 Stack Edition

Ограничение по времени: 1 second

Ограничение по памяти: 512 mebibytes

Недавно девочка Ксюша купила в онлайн магазине приложение «1024 Stack Edition» — пошаговую аркаду. Как можно догадаться из названия разворачивается вокруг некоторого стека, наполненного степенями двойки. По правилам игры, на каждом ходу игрок может выполнить одно из трех действий над стеком:

- Добавить случайное число в конец стека.
- Убрать последнее число из стека.
- Если два последних числа равны, то заменить их на их сумму.

Если игрок выбирает добавить случайное число в стек, то с шансом p в стек добавится число 1 и с шансом $1-p$ добавится число 2. Если игрок выбирает убрать последнее число из стека, он платит штраф в 1 монету. Обратите внимание, что все остальные операции, кроме удаления последнего элемента из стека, не приводят к уплате штрафа. Для того чтобы выиграть, игроку необходимо привести стек в такое состояние, в нем находится лишь один элемент — 1024. При этом, чем меньше штрафа он заплатит — тем лучше.

Как-то раз Ксюша оставила свой телефон без присмотра, а по возвращению обнаружила, что кто-то играл в «1024 Stack Edition». Ксюше не интересно узнать, кто это был и каковы были его мотивы. Ей интересно узнать, каково математическое ожидание штрафа, который лично ей придется заплатить при оптимальной игре, если она доиграет партию. Ваша задача, как вы могли уже догадаться, посчитать это число.

Разбор задачи А

Для начала рассмотрим случай, когда $N = 0$. Пусть $f(i)$ — это ожидаемое наименьшее количество монет, которое потребуется потратить, чтобы собрать число 2^i . Приведем формулы для подсчета $f(i)$ в случае, если $p < 100$. Если же $p = 100$, то эти же формулы адаптируются тривиальным образом.

1. $f(0) = 1.0/p - 1$;
2. $f(1) = \min(f(0) \cdot 2; 1/(1-p) - 1, 1-p)$;
3. $f(i) = 2 \cdot f(i-1), i > 1$.

В случае с $N = 0$ ответом на задачу будет $f(10)$.

Теперь предположим, что $N > 0$. Обозначим через $g(i, j)$ ожидаемое наименьшее количество монет, которые нам нужно будет потратить, чтобы последних (верхних) i элементов стека получить только одно число 2^j . Подсчет $g(i, *)$ для фиксированного i будет происходить в 2 этапа (где $S(i)$ обозначим двоичный логарифм i -го сверху элемента стека):

1. Посчитаем $g(i, S(i)) = 1 + \min(g(i-1, *)), i > 1$. Этот случай соответствует тому, что из всего, что выше элемента i , мы получаем некоторое число, после чего удаляем его.

Также посчитаем $g(i, S(i) + 1) = g(i-1, S(i))$. Этот случай соответствует тому, что мы из всех элементов, расположенных выше i , собираем $2^{S(i)}$, после чего объединяем его с текущим элементом, который также равен $2^{S(i)}$. Для всех j , отличных от $S(i)$ и $S(i) + 1$, положим $g(i, j) = +\infty$.

2. На первом этапе мы «разобрались» с числом $S(i)$. Прежде чем переходить к $g(i+1, *)$ и числу $S(i+1)$, мы можем произвести некоторые действия с i -м элементом стека, превратив его в любой другой элемент. Чтобы это учесть в динамике, пересчитаем $g(i, *)$ следующим образом:

- $g(i, 0) = \min(g(i, 0), \min(g(i, *)) + 1.0/p)$ — мы либо используем уже найденное значение $g(i, 0)$, либо берем любое число, удаляем его и с туда $1 = 2^0$.
- $g(i, j) = \min(g(i, j), \min(g(i, *)) + 1 + f(j), g(i, j-1) + f(j-1)), j > 0$. Здесь мы либо удаляем стоявшее на месте i число и затем за $f(j)$ монет ст туда число 2^j , либо ставим за $g(i, j-1)$ шагов число 2^{j-1} , собираем за $f(j-1)$ шагов еще 2^{j-1} и объединяем их.

Ответом на задачу будет $g(N, 10)$. Следует обратить внимание, что в промежуточных вычислениях следует также считать $g(*, 11)$. Сложность решения $O(NR)$, где R — максимальная степень двойки, которая присутствует во входных данных (в данном случае $R \leq 10$).



Задача В. Строчная задача

Ограничение по времени: 4 seconds

Ограничение по памяти: 512 mebibytes

Поисковые запросы — что может быть лучше? Поле для ввода вопроса, кнопка «найти» — и весь интернет выстраивается в упорядоченный список ответов. Миллионы поисковых запросов сохраняются каждый день, чтобы в будущем отвечать на них быстрее и лучше, но этого мало!

Важно помочь пользователю быстрее вводить сам поисковый запрос. Например, можно подсказывать популярные или интересные пользовательские варианты запросов, упорядочивая их в соответствии с предпочтениями. Но что делать, если мы ничего не знаем о том, кто задаёт вопрос? Вам поручено разработать алгоритм, который будет предлагать две подсказки в лексикографическом порядке. Ну, почти в лексикографическом.

Известно, что пользователи могут допустить опечатки при вводе запроса. Скажем, что строка A почти меньше строки B , если в строке A можно заменить не более одной буквы на произвольную маленькую букву латинского алфавита так, чтобы строка A стала лексикографически меньше строки B .

Вам даны N слов-подсказок. Посчитайте количество пар слов, первое из которых почти меньше второго. Количество пар нужно вычислять с учётом порядка: если два слова являются взаимно почти меньшими, то учитывайте обе пары.

Разбор задачи В

Рассмотрим пару строк s и w . Строка s почти меньше строки w тогда и только тогда, когда минимальная лексикографически строка $s^{\min}$, которую можно получить из s заменой одной буквы, строго меньше w .

Однако, как легко видеть, $s^{\min}$ получается из строки s заменой первого символа не равного 'a' на символ 'a'. Если строка s состоит только из символов 'a', то $s = s^{\min}$. Рассмотрим строку s_i , и найдем C_i — количество строк s_j (возможно $i = j$) таких, что $s^{\min}_i < s_j$. Тогда к ответу нужно прибавить $C_i - 1$, если $s^{\min}_i \neq s_i$, и C_i в противном случае.

Для нахождения величин C_i можно воспользоваться двумя способами:

- отсортировать строки s_i и находить C_i двоичным поиском;
- добавить все строки в **trie**, и находить C_i спуском по нему.

Задача С. Кубик

Ограничение по времени: 2 seconds

Ограничение по памяти: 512 mebibytes

Дан куб, грани которого пронумерованы различными целыми числами от 1 до 6. Также дано гексамино, все клетки которого пронумерованы различными целыми числами от 1 до 6.

Определите, можно ли из заданного гексамино сделать куб, который можно повернуть так, чтобы нумерация граней совпала с нумерацией π заданного куба, если разрешается только сгибать гексамино по границам клеток, но не разрезать его.

Разбор задачи C

Основное, что надо заметить в этой задаче — гексамино можно сгибать в разные стороны. При этом получаются два кубика, отличающихся друг от друга только взаимной перестановкой одной пары противоположных цифр (так как поворот на 180 градусов переставляет две пары противоположных цифр, то перестановка всех трёх пар эквивалентна перестановке одной пары).

Далее существует несколько методов решения. Один из них состоит из следующих шагов:

1. Устанавливаем кубик нижней стороной на соответствующую клетку гексамино.
2. Для каждого из четырёх поворотов кубика вокруг своей оси начинаем перекачивать его по гексамино до тех пор, пока возможно перекачать кубик так, чтобы нижняя грань попала на клетку с соответствующим номером.
3. Если для какого-то из поворотов все 6 клеток посещены, то ответ "Yes".
4. Если нет, то меняем местами цифры на правой и левой гранях и повторяем процедуру с шага 2.
5. Если и в этом случае ни при одном из поворотов не посещены все 6 клеток, то ответ — "No".

Также существует возможность предприсчитать все 11 развёрток и затем проверять совпадение гексамино (поворачивая его в стандартное положение), а в случае совпадения — соответствие пар чисел, использованных для нумерации противоположных сторон кубика, с теми же парами в развёртке.



Задача D. НЛО

Ограничение по времени: 1 second

Ограничение по памяти: 256 mebibytes

Ночью в озеро, расположенное неподалеку от Васиного дома, упал НЛО. НЛО представлял собой каркас N -мерного прямоугольного параллелепипеда, собранный из титановых рёбер, причём все рёбра имели целые длины. В момент крушения соединения между рёбрами разрушились, в отличие от самих рёбер, которые оказались на дне и на берегу озера.

С утра Вася пришёл на берег озера и нашёл K титановых стержней. Предполагая, что эти стержни являются рёбрами каркаса НЛО и что, воз часть рёбер ещё находится под водой, определите наименьшую возможную размерность пространства, из которого прибыл НЛО.

Решение задачи D

Это самая простая задача набора. Очевидно, что размерность параллелепипеда не меньше количества различных длин рёбер. Также замети K -мерном параллелепипеде 2^{K-1} рёбер одного типа, то есть в случае наличия X одинаковых рёбер для них будет использовано $\lceil (X-1)/2^k \rceil$ измерение.

Поэтому в порядке возрастания перебираем все размерности, пока вычисленное по указанной выше формуле суммарное количество использованных измерений не будет меньше или равно текущей размерности. Так как при $K = 21$ рёбер одного типа будет $2^{20} > 10^6$, то пер будет небольшим.

Задача E. Недополняемое паросочетание

Ограничение по времени: 1 second

Ограничение по памяти: 512 mebibytes

Артёмка очень любит паросочетания. Паросочетание в неориентированном графе — это набор попарно несмежных ребер. Паросочетание называется недополняемым, если невозможно добавить в паросочетание ещё одно ребро, не удаляя при этом ни одно из уже включённых рёбер.

Неориентированный граф называется двудольным, если его вершины можно разбить на два множества так, чтобы любое ребро графа соединяло вершины из разных множеств.

Артёмке дан двудольный граф. Его задача — найти количество недополняемых паросочетаний в нем по модулю $1\,000\,000\,007 (10^9 + 7)$. Ваш задача — помочь ему.

Решение задачи E

Для решения задачи заметим, что число вершин в левой доле невелико.

Будем считать следующую динамику, добавляя по одной вершине из правой доли: $f(i, j)$ — количество способов расставить рёбра в случае, если мы рассмотрели первые i вершин правой доли, а вершины левой доли заданы троичной маской j следующим образом: 0 — вершина не могла быть соединена ребром ни с одной вершиной из правой доли, 1 — вершина могла быть соединена ребром ни с одной вершиной из правой доли, 2 — вершина соединена ребром с одной из вершин правой доли.

Таким образом, рассматривая очередную вершину из правой доли, мы можем пересчитать маску для вершин левой доли. Очевидно, что при подсчете ответа нужно рассматривать маски без 1.

Таким образом, ответом будет сумма $f(n_2, \text{ValidMask})$, для всех троичных масок ValidMask , не содержащих единиц в троичной записи.



Задача F. Музыкальный мир

Ограничение по времени: 1 second

Ограничение по памяти: 512 mebibytes

В обновленной Яндекс.Музыке музыкальные рекомендации стали ещё лучше, чем раньше! Они учитывают каждый прослушанный пользователем трек с самого начала использования сервиса, и в зависимости от уже прослушанных — увеличивают пул рекомендаций. Плеер включает пользователю случайный из рекомендованных треков, и так целый день.

Но сервера Яндекса такие быстрые, что могут строить рекомендации не только для того трека, которую сейчас слушает пользователь, но для вариантов случайно выбранных следующих треков.

Пользователь выбирает первый трек, а дальше алгоритм автоматически строит рекомендации на N песен вперёд. Из каждой песни i -го поколения может получиться не более K_i ($1 \leq K_i \leq 5$) новых песен в $(i + 1)$ -м поколении, причём для i -го поколения известна вероятность p_{ij} , j того, что основании песни i -го поколения будет предсказано j новых песен для каждого j от 0 до K_i .

Так как экспериментальное исследование производилось на жанре инди, все рекомендованные треки различны. Для того чтобы измерить разнообразие сбора статистики о похожести песен, похожесть измеряется для каждой пары треков. Чтобы помочь разработчикам Яндекса оц затраты по памяти для этого измерения, вычислите математическое ожидание количества пар треков в пуле рекомендаций в $(N + 1)$ -м поколении.

Решение задачи F

Обозначим за Y_i случайную величину, равную количеству песен в i -ом поколении. Предлагаемое решение задачи основано на понятии производящей функции. Если вы не знакомы с ним, ознакомиться можно, к примеру, в статье Википедии ru.wikipedia.org/wiki/Производящая_функция_последовательности.

Рассмотрим производящие функции заданных во входе случайных величин $\psi_i(x) = \sum_{j=0}^{K_i} p_{ij} x^j$, а

также производящие функции случайных величин $\varphi_i(x) = \sum_{j=0}^{\infty} \mathbb{P}\{Y_i = j\} x^j$.

У нас есть явный вид всех ψ_i , а также явный вид $\varphi_1(x) = x$. Научимся выражать φ_{i+1} через ψ_i и φ_i .

В случае, если бы мы точно знали, что $Y_i = k$, то можно было бы утверждать, что $\varphi_{i+1}(x) = \psi_i(x)^k$. Но так как Y_i — случайная величина, то ис производящая функция есть сумма производящих функций, соответствующих разным значениям, с весами, равными вероятностям этих знач

$$\text{то есть } \sum_{j=0}^{\infty} \mathbb{P}\{Y_i = j\} \psi_i(x)^j = \varphi_i(\psi_i(x)).$$

Так как максимальное значение искомой величины очень большое, вычислить явный вид производящей функции для Y_{n+1} не представляется возможным. Однако он нам и не нужен. Несложно заметить, что величина, которую необходимо посчитать

$$\mathbb{E} \frac{Y_i(Y_i - 1)}{2} = \sum_{j=0}^{\infty} \mathbb{P}\{Y_i = j\} \frac{j(j-1)}{2} = \frac{1}{2} \varphi_i''(1)$$

Научимся пересчитывать значение, первую и вторую производную функции $\varphi_{i+1}(x)$ в (1) через предыдущие функции. Напомню, что $\varphi_i(1) = 1$. Как мы знаем, $\varphi_{i+1}(x) = \varphi_i(\psi_i(x))$. Тогда

$$\begin{aligned} \varphi'_{i+1}(x) &= \varphi'_i(\psi_i(x)) \psi'_i(x) \\ \varphi''_{i+1}(x) &= \varphi''_i(\psi_i(x)) \psi'_i(x)^2 + \varphi'_i(\psi_i(x)) \psi''_i(x) \end{aligned}$$

При подстановке $x = 1$ эти формулы упрощаются и принимают вид

$$\begin{aligned} \varphi'_{i+1}(1) &= \varphi'_i(1) \psi'_i(1) \\ \varphi''_{i+1}(1) &= \varphi''_i(1) \psi'_i(1)^2 + \varphi'_i(1) \psi''_i(1) \end{aligned}$$

Также несложно понять, что вычисления можно проводить сразу по модулю $10^9 + 7$, заменив исходные вероятности на соответствующие значения по модулю.

яндекс.алгоритм, разбор задач, геннадий короткевич

↑ +67 ↓

👁 52,8k ⭐ 270

🐦 📧 📧

👤

Автор: @lperovskaya

Я

рейтинг

Яндекс 795,02

Как мы делаем Яндекс

Github

ПОХОЖИЕ ПУБЛИКАЦИИ

22 августа 2013 в 19:16

Разбор всех задач и результаты Яндекс.Алгоритма

↑ +71 👁 94,6k ⭐ 384 💬 30

12 июля 2013 в 12:08

Разбор задач финала чемпионата мира по программированию ACM ICPC 2013

↑ +107 👁 88,5k ⭐ 384 💬 13

21 июня 2013 в 14:42

Яндекс.Алгоритм 2013: новая платформа Яндекс.Contest и правила TCM/Time

↑ +37 👁 11,6k ⭐ 43 💬 20

Комментарии (32)

ewgRa 4 августа 2014 в 16:18 #

+1

Эх, мне решения можно в качестве задач давать, недельки на три наверное хватит.

Интересно взять статистику семейного положения, возраста, трудоустроенности участников таких соревнований. Если там большой пик на холостой молодежи можно утешать себя мыслью, что если бы не семья-дети-работа, то вот я бы...

Победившим поздравления!

nochkin 4 августа 2014 в 19:20 #

+

https://habrahabr.ru/company/yandex/blog/231939/

7/12

А если у них семья, то можно утешать мыслью, что оптимизация мытья тарелок после ужина тоже является ещё той задачкой.

А если у них есть дети, то можно утешать мыслью, что наше подрастающее поколение настолько продвинуто, то может учить своих родителей. Но ведь это научили их мы.

А если у них работа, то можно утешать мыслью, что при сложных условиях работы в невесомости накладываются иные взгляды на решаемые проблемы.

А если у них...

 cigan 4 августа 2014 в 17:05 #

А что за «штрафное время»? Плохо стрелял в CS? :)

Прочёл все задачи и не понял одного — а причём тут программирование вообще? Ну отбери у них компьютеры и дай карандаш с бумагой — ТОЧНО ТАК ЖЕ Е МОЖНО РЕШИТЬ. Зачем тогда тут приплетать компы? Это тервер, комбинаторика, матан — первые три курса института. /хотя каюсь, ничего из этого я уже не забыл :)/

 Iperovskaya 4 августа 2014 в 17:30 # h ↑

Штрафное время складывается из штрафа за неверную попытку (20 штрафных минут) и времени, потраченного на решение задачи (например, 40 штрафных минут, если задача была успешно решена на сороковой минуте). Кроме того, можно получить за задачу отрицательное штрафное время, отправив её без права на попытки.

Кроме математики и теории вероятностей в задачах используются и структуры данных, значение имеет эффективность их использования. Именно умение не только решить задачу математически, но и верно реализовать решение отличает программирование от математики.

 cigan 5 августа 2014 в 02:23 # h ↑

Это не программирование, а математика + обсчёт на компьютере. Весь пост — сплошные формулы, что здесь от программирования? Классы? Шаблон проектирования? Биты/байты? Или вы «программированием» уже готовы назвать суммирование в цикле?

Я не спорю, писать программы надо уметь, но с таким успехом и плотник, рассчитавший гипотенузу в Икселе — программист! Вам самой не смешно называть это программированием?

PS

Про «штрафы» — шутка же, совсем с работой заморочились? :) CS=Counter Strike, намёк на биатлон

 Larick 6 августа 2014 в 14:31 # h ↑

Классы/шаблоны/etc — не более, чем инструмент. Задача программиста как раз в том, чтобы уметь спроецировать реальный мир на математические абстракции, описать реальный мир языком цифр, а именно с цифрами и работает компьютер.

А конечным инструментом может оказаться процедурное/ООП/функциональное программирование, это уже не так важно в данном контексте.

Приведу пусть и не точную, но (мне так кажется) уместную аналогию: на конкурсе мастеров деревообработки не станут проверять мастерство работы станком, а как раз выдадут рубанок/молоток/нож и так далее.

 cigan 6 августа 2014 в 19:37 # h ↑

Ясно. Понимание программирования на уровне «Если в работа кидаться макаронами, то он сломается, потому что от них у него защиты нет».

 Olorin 4 августа 2014 в 17:59 # h ↑

Смысл конкурса в том, чтобы определить, кто лучший программист, а смысл программиста — в том, чтобы учить компьютер наиболее оптимально решать задачи. Всё логично.

 nochkin 4 августа 2014 в 19:31 # h ↑

Тут задачки конкретно на программирование и оптимальные алгоритмы, для которых нужны знания математики и смежных наук, а не наоборот.

Ведь, грубо говоря, почти любую компьютерную задачу можно решить разными способами:

1. Напрямом, перебор и т.п. Это займёт уйму времени (может даже и жизни не хватить)
2. Оптимальный. Тут на подмогу приходят знания алгоритмов, умения правильно пользоваться структурами данных и знать их слабые и сильные стороны и благодаря этому долгую задачу из #1 можно сократить до порядка минут или секунд.

Особенно это хорошо чувствуется когда идёт обработка больших массивов данных, так как, например, разные алгоритмы эффективны с разным объёмом данных. Так называемое «решение в железе», которое можно применить в реальных условиях и летать с ними на луну.

А на бумаге это больше как теоретическое решение, а не практическое. То же можно решить, но это другие цели.

 cigan 8 августа 2014 в 03:09 # h ↑

Ну да, всё верно.

«Нам нужно нарисовать борщ, для этого воспользуемся картинкой свёклы и напишем программу, выводющую в случайных местах нашу свёклу» — вот так специалист по борщам я сейчас вижу, но никак не программиста.

 nochkin 8 августа 2014 в 03:32 # h ↑

Примерно так, но только надо это сделать оптимальнее, что бы, например, свёкла не выводилась два раза в одном и том же месте и что бы учитывать нахождение других овощей.

 cigan 8 августа 2014 в 05:00 # h ↑

Ну вот! Уже ближе к программированию. Осталось только закрепить мысль:

Программирование ДАВНО уже перестало быть частью «математики» и стало абсолютно самостоятельной областью, со своими методами, концепт и «аксиомами». Математика, физика, ДАЖЕ ЭСТЕТИКА стали на службе ИТ, но это никак не превращает программирование в решение задач для ИТ. Эта мысль не вызывает споров? Равно как и мысль, что 99% задач программистов даже рядом не стоят с математическими абстракциями.

Отсюда, возвращаемся к изначальному посту: что конкретно компьютерного вы нашли в задачах от тындекса? Они что, общаются с СУБД? Обрабатывают прерывания? По сетям что-то передают? Достаточно глянуть на решения — это же сплошь математика и формулы, какой им компьютеры-то? Про то что математик может закодировать цикл? Так это уже вопрос квалификации МАТЕМАТИКА, но никак не задача для профессиональных программистов. Если я вам скажу, что есть такая компьютерная должность «system analyst», вы со стыда сгорите — это как раз тот человек, который избавляет программистов вообще от какой-либо математики/бухгалтерии/физики, превращая требования реального мира в чисто компьютерное ТЗ (надеюсь надо расшифровывать?). Верьте мне, я — программист и знаю лучше дамочек-пиарщиц чем мы занимаемся на работе. :)



nochkin 8 августа 2014 в 05:13 # h ↑

К сожалению, не согласен. Не знаю почему я похож на «дамочку-пиарщицу», но это не так важно.

Математика нужна для современного программирования настолько, что некоторые вполне рядовые задачи просто не решить без этих знаний. Например, обработка звука и изображений, анализ речи, компрессия потока данных в реальном времени и т.п.

Закодировать цикл может любой, а вот что бы закодировать цикл так, что бы он использовал меньше ресурсов и выполнялся быстрее — это уж другое. Тут надо начинать думать иначе.



cigan 8 августа 2014 в 14:33 # h ↑

> Не знаю почему я похож на «дамочку-пиарщицу», но это не так важно.

:)))))))))) Это важно, потому что речь шла о Iperovskaya, с которой и началась эта довольно глупая дискуссия «математика и программирование». Вы это восприняли на себя ввиду излишней занудности и заикленности — грустный побочный эффект увлечения математикой. :)

> Математика нужна...

Зачем писать банальщину? Вы считаете, что я за 15 лет п-я этого не понял? Это _Я_ вам объясняю, что БОЛЬШИНСТВО п-ских задач даже не стоят с математикой, т.к. решают сугубо компьютерные проблемы — пересылка данных, оптимизации, удобство интерфейса и т.п. Вы носите в объявлениях (ну вот те, которые из реальной жизни) — много там «математики»? Сплошь «похапасты», «облака», «геймдев» и прочее. Чему мне ваши увещания в математике, месье? Вы совсем не понимаете, что объясняете проф.программисту чем он оказывается занимается. Я-то каждый день пишу программы, с людьми разговариваю — мне лучше знать процент вовлечения других дисциплин.

Вот вам сайт: rstdn.ru слэш forum/dotnet/ — сходите что ли, посмотрите чем занимаются люди, чтоб ерунду не писать.



Zalina 8 августа 2014 в 14:57 # h ↑

Лидя не дамочка-пиарщица, а выпускница ИТМО.



cigan 8 августа 2014 в 16:45 # h ↑

Разве одно другому мешает? Пол Москвы — «выпускники» чего-либо, правда работают, прости господи, вплоть до «ленинградки» :) Хотя речь не о том, а о том, что человек, работающий в Яндексе, не имеет никакого понимания чем занимается программист. Весь этот про математику — умильтельная, но далёкая от реальности идея. Выше я уже дал ссылку на сайт — почитайте их вопросы (по реальной работе) и проверьте, сколько там математики.



Mrri 8 августа 2014 в 18:41 # h ↑

Сами же пишете — «99% задач программистов даже рядом не стоят с математическими абстракциями». Следовательно, каждая десятая задача каждого десятого программиста (условно) *требует* владения математикой. На абстрактном уровне. И когда этот программист столкнётся с этой задачей — он просто не справится со своей работой. 1% — это огромная величина, особенно при современных объемах программирования.



cigan 8 августа 2014 в 22:00 # h ↑

Требует, но не от программиста. Почитайте кто и чем занимается в составе девелоперской команды:

Business Analyst
Application Architect
DBA
Programmer
Tester
UX specialist
Designer

Все эти люди «как бы пишут программу», но каждый занимается своим делом. Говорить, что программисту нужно хорошо знать математику — не менее тупо, чем требовать от повара список удобрений для морковки — это НЕ ЕГО РАБОТА.

Это в России привыкли на одном Ване пахать за семерых — отсюда и заблуждение.



Mrri 8 августа 2014 в 23:32 # h ↑

На этой неделе я нарисовал граф обработки данных в нашем проекте. Каждая из десятка с лишним функций, которые попали была математической задачей, на которую я потратил по несколько месяцев. Причём там была и разработка алгоритма, и кодирование, и много экспериментов. Предсказать заранее, каким должно быть разделение на блоки, и что у них на входе/выходе мне не удавалось. Как можно было бы реализовать это дело, если бы я только разрабатывал алгоритмы и передавал их коды представляю. Вообще.



cigan 9 августа 2014 в 00:58 # h ↑

И что? А у Васи Иванова вообще нет цифр и формул в проекте. Что будем делать? Настырно проталкивать мысль о важности математики или всё же сходим на RSDN? Пока вы там не прочтёте хотя бы 100 вопросов, не вижу смысла тут «я-кать» со графами.



Mrri1 9 августа 2014 в 11:16 # h t

Если у Васи в проекте нет формул, то с большой вероятностью ему не приходится реализовывать запросы математика, кс пытается перевести свои заумные идеи на понятный Васе язык. Так что речь вообще не о нём. Вот если бы формулы были. Вася при этом не знал математики, то тогда бы мы посмотрели, как они будут общаться и к чему это приведёт — ведь у каждого своя задача, и математика к программированию не пускают, не так ли?



nochkin 8 августа 2014 в 22:02 # h t

Объявления не являются показателем всей сферы.

То нанимают «эникейщиков» (так вроде это называется). То есть, люди, которые чисто механически умеют жать на кнопки в правильной последовательности. Для этого много знаний не нужно и, конечно, результат будет соответствующий. После этого эти самые «похапастые» проекты работают тят-ляп, жрут неимоверное количество ресурсов и не могут решить простые задачи достаточно быстро. После этого уже начинаются поиски более оптимальных алгоритмов, моделей и прочего. Или проект просто закрывают.

Эти объявления совсем не означают, что задачи программистов не стоят с теориями, алгоритмами и прочими моделями. Ведь есть нормальные компании, которые иначе подходят к проблеме и решают их более грамотно. Особенно сегодня когда объёмы данных растут быстрее, чем вчера.

И это объясняю я, как не математик.



andy1618 8 августа 2014 в 05:33 # h t

А если я вам скажу, что есть такая компьютерная должность «system analyst», вы со стыда сгорите — это как раз тот человек, который избавляет программистов вообще от какой-либо математики/бухгалтерии/физики, превращая требования реального мира в чисто компьютерное ТЗ

:)

Хм, а Вы не путаете «программиста» и «кодера»?



Larick 8 августа 2014 в 11:08 # h t

Как насчёт калибровки энкодера в программе? и позиционирования с учетом люфта в механизмах? в ТЗ ни одна сволочь не посчитает нужным описать (чаще всего именно так, и, вероятно, потому что сами не представляют этого). А Вам, как программисту, все-равно придется это реализовывать.

2. Понадобилось рассчитать угол отклонения вала рулевого колеса для прохождения заданного поворота на заданной скорости по заданному радиусу и грунту. Поверьте, математик разберется в программировании, но тогда уже вы окажетесь не нужны



Mrri1 8 августа 2014 в 18:49 # h t

Калибровка энкодера — тоже интересно. Особенно если в реальном времени (интересно, на основании каких данных). А ещё и рассчитывать управление мотором, чтобы обеспечить равномерное движение на всём участке и остановку в нужном положении (с точностью хотя бы 10-1 секунда)... Теория там не помощник, нужен анализ эмпирических данных.



andy1618 8 августа 2014 в 23:01 # h t

Теория там не помощник...

Мы тоже как-то без теории написали программу для управления системой с обратной связью. Всё бы ничего, но примерно раз в неделю с началом дико «колбасить», т.е. возникал очень сильный автоколебательный процесс. Пришлось таки брать за теорию и курить критерии устойчивости Найквиста и т.п. — после этого всё заработало как часы!



andy1618 5 августа 2014 в 01:44 #

Дан куб, грани которого пронумерованы различными целыми числами от 1 до 6. Также дано гексамино, все клетки которого пронумерованы различными целыми числами от 1 до 6. Определите,...

Странно, что в условии не поясняется, что такое «гексамино».

К примеру, в гугл-турнирах мало того, что всё тщательно разжуют по два раза на уровне «домохозяйек», так ещё и кучу картинок красивых нарисуют, чтобы искоренить неоднозначности. А тут сухая жёсть какая-то.



andy1618 8 августа 2014 в 05:36 (комментарий был изменён) # h t

UPD: Почитал оригиналы условий — вопросы отпали. Там всё объясняется, есть примеры входных/выходных данных и ограничения на них, что тоже очень помогает для осознания масштаба проблемы :)

Хотя картинок не хватает — факт!



deleted-lucky_brick 8 августа 2014 в 23:17 (комментарий был изменён) #

те кто нихрена не понимает в математике пишет — что математика в голом виде не программирование и ну и конечно отчасти обидно, что ты типо лох! а эти качки-стероидные они то могут херануть интеграл в уме!

в сущности повседневный спектр задач — это CRUD-technology =)
но если будете строить рекомендательную систему то пару формул выучить придется.

те кто нихрена не понимает в программировании (математики тобишь) — считают что методологии программирования на которых щас все и держится это все и
надо бы уметь решать алгоритмы на бумажке за 3 минуты.

видали мы код таких «математиков»

кстати слышали мы, что в яндексе мало платят ;)

 **Kongo** 25 августа 2014 в 17:35 #

Поможет ли победа в Тур де Франс стать самым коммерчески успешным токийским рикшей? Вроде бы занятие технически весьма близкое.

 **Mrrl** 26 августа 2014 в 14:29 # h ↑

Может стать помочь каким-нибудь велокурьером экстра-класса. И это может быть для победителя интереснее, чем работа рикшей.

 **Kongo** 28 августа 2014 в 12:27 # h ↑

Именно. Это разные направления саморазвития. Зачем их сравнивать?

Только зарегистрированные пользователи могут оставлять комментарии. [Войдите](#), пожалуйста.

САМОЕ ЧИТАЕМОЕ

Разра

Сейчас

Сутки

Неделя

Месяц

Анализ шифровальщика Wana Decrypt0r 2.0

+96

37,3k

124

225

О том, как в Instagram отключили сборщик мусора Python и начали жить

+26

6,4k

59

3

Как защищаться от атаки вируса-шифровальщика «WannaCry»

+2

9,9k

38

22

Создание JPEG из ниоткуда

+35

7,6k

30

2

Познакомимся с WannaCry поближе

+45

38,2k

87

61

ИНТЕРЕСНЫЕ ПУБЛИКАЦИИ

Гигер и фантастические технологии мира Чужих

1,5k

6

2

GT

Колония. Глава 11: Рассказ доктора

333

0

3

GT

Физики впервые установили контрфактическое квантовое соединение

1,7k

8

6

GT

Джефф Безос: будущее бизнеса — машинное обучение, концентрация на интересах клиентов и быстрое принятие решений

742

2

0

GT

Да будет фильм с Xamarin.Forms

Аккаунт	Разделы	Информация	Услуги	Приложения
Войти	Публикации	О сайте	Реклама	 Загрузите в App Store  доступно Google
Регистрация	Хабы	Правила	Тарифы	
	Компании	Помощь	Контент	
	Пользователи	Соглашение	Семинары	
	Песочница	Помощь стартапам		
 © 2006 – 2017 «ТМ»		Служба поддержки	Мобильная версия	