Задачки деда. Старый хитрее, стырый мудрее.

ОПГ Овчинников

20 октября 2023 г.

Артемий

1. **Асимптотика. Обозначения o, $\mathcal{O}$, Θ, Ω, ω.**
Асимптотика - поведение функции при стремлении аргумента к бесконечности.
Пусть $f(n)$ и $g(n)$ - две функции, которые стремяться к бесконечности, тогда:

- $f(n) = o(g(n))$, если g доминирует над f асимптотически, то есть $\forall(\mathrm{C} > 0)$ $\exists N : \forall(n \in N)\ |f(n)| < C|g(n)|$
- $f(n) = \mathcal{O}(g(n))$, если f ограничена сверху функцией g асимптотически, то есть $\forall(\mathrm{C} > 0)\ \exists N : \forall(n \in N)\ |f(n)| \leq C|g(n)|$
- $f(n) = \Theta(g(n))$, если f ограничена снизу и сверху функцией g асимптотически, то есть $\forall(\mathrm{C}_1 > 0), (C_2 > 0)\ \exists N : \forall(n \in N)\ C_1|g(n)| \leq |f(n)| \leq C_2|g(n)|$
- $f(n) = \Omega(g(n))$, если f ограничена снизу функцией g асимптотически, то есть $\forall(\mathrm{C} > 0)\ \exists N : \forall(n \in N)\ |f(n)| \geq C|g(n)|$
- $f(n) = \omega(g(n))$, если f доминирует над g асимптотически, то есть $\forall(\mathrm{C} > 0)$ $\exists N : \forall(n \in N)\ |f(n)| > C|g(n)|$

Прекрасная лекция на 30-40 минут, в которой эти штуки объясняют в том числе графически, что намного легче для понимания, даже необязательно всё смотреть: *ТЫК*.

2. **Основные свойства асимптотики. Асимптотика многочлена.**
Тут если первый билет понять, проблем быть не должно.
Транзитивность:

- $f(n) = \Theta(g(n)) \land g(n) = \Theta(h(n)) \Rightarrow f(n) = \Theta(h(n))$
- $f(n) = \mathcal{O}(g(n)) \land g(n) = \mathcal{O}(h(n)) \Rightarrow f(n) = \mathcal{O}(h(n))$

- $f(n) = \Omega(g(n)) \land g(n) = \Omega(h(n)) \Rightarrow f(n) = \Omega(h(n))$
- $f(n) = o(g(n)) \land g(n) = o(h(n)) \Rightarrow f(n) = o(h(n))$
- $f(n) = \omega(g(n)) \land g(n) = \omega(h(n)) \Rightarrow f(n) = \omega(h(n))$

Рефлексивность:
$f(n) = \Theta(g(n)) \Rightarrow f(n) = \mathcal{O}(g(n))$; $f(n) = \Omega(g(n))$
Симметричность:
$f(n) = \Theta(g(n)) \Rightarrow g(n) = \Theta(f(n))$
Перестановочная симметрия:
$f(n) = \mathcal{O}(g(n)) \Leftrightarrow g(n) = \Omega(f(n))$
$f(n) = o(g(n)) \Leftrightarrow g(n) = \omega(f(n))$
Хз, на вики нет названия))):

- $C \cdot o(f(n)) = o(f(n))$
  $C \cdot \mathcal{O}(f(n)) = \mathcal{O}(f)$
- $o(C \cdot f) = o(f)$
  $\mathcal{O}(C \cdot f) = \mathcal{O}(f)$
- $o(-f) = o(f)$
  $\mathcal{O}(-f) = \mathcal{O}(f)$
- $o(f) + o(f) = o(f)$
  $o(f) + \mathcal{O}(f) = \mathcal{O}(f) + \mathcal{O}(f) = \mathcal{O}(f)$
- $\mathcal{O}(f) \cdot \mathcal{O}(g) = \mathcal{O}(fg)$
  $o(f) \cdot \mathcal{O}(g) = o(f) \cdot o(f) = o(fg)$
- $\mathcal{O}(\mathcal{O}(f)) = \mathcal{O}(f)$
  $o(o(f) = o(\mathcal{O}(f)) = \mathcal{O}(o(f)) = o(f)$

Асимтотика многочлена:
$P(x)$ - многочлен, тогда $P(x) = \Theta(x^{degP})$ при старшем коэффициенте > 0.
$P(x) = a_0 + a_1x + a_2x^2 + ... + a_kx^k$, все слагаемые кроме a_kx^k являются $o(x^{degP})$.
Значит вся сумма является $\Theta(x^k)$.

3. **Определение o, ω, $\mathcal{O}$ через пределы.**
$f = o(g)$: $\lim\limits_{n\to+\infty} \dfrac{f(n)}{g(n)} = 0$

$f = \omega(g)$: $\lim\limits_{n\to+\infty} \dfrac{g(n)}{f(n)} = 0$

$f = \mathcal{O}(g)$: $\overline{\lim\limits_{n\to+\infty}} \dfrac{f(n)}{g(n)} < \infty$

4. **Отношение доминирования между основными функциями.**
Если я правильно понял, то вот:
$\log\log n < \log n < n < n\log n < n^2 < n^3 < 2^n < n! < n^n$

5-6. **Определение однородных линейных рекуррентных соотношений (ОЛРУ). Алгоритм поиска решения. Случай разных корней характеристического уравнения. Случай кратных корней характеристического уравнения.**
Определение:
Последовательность $u_1, u_2, ..., u_n, ...$ называется однородной линейной рекуррентной последовательностью порядка k, если существуют натуральное число k и числа $a_1, a_2, ..., a_k$ (действительные или комплексные, причем $a_k \neq 0$) такие, что для любого n справедливо $u_{n+k} = a_1 u_{n+k-1} + a_2 u_{n+k-2} + ... + a_k u_n$ ($\leftarrow$ это и называется однородным линейным рекуррентным соотношением порядка k).
Алгоритм решения (далее будут примеры, на которых понятнее):
$f_{n-k} = a_1 f_{n+k-1} + ... + a_k f_n$
$r^k = a_1 r^{k-1} + ... + a_k$
$r^k - a_1 r^{k-1} - ... - a_k = 0$
Далее находим корни получившегося уравнения и считаем кратность L каждого корня
Общее решение:
$f_{n+k} = S_1 + S_2 + ... + S_k$, причём
$S_i = (c_{i1} + c_{i2} n + c_{i3} n^2 + ... + c_{iL} n^{L-1}) \cdot r_i^n$
$r_i^n \leftarrow$ сам корень
Ищем частное решение (получаем k первых элементов и составляем систему):
$f_0 = v_0$, $f_1 = v_1$, ..., $f_k = v_k$

$$\begin{cases} \mathrm{v}_0 = \sum_{j=1}^{k} S_j \\ \mathrm{v}_1 = \sum_{j=1}^{k} S_j \\ ... \\ \mathrm{v}_k = \sum_{j=1}^{k} S_j \end{cases}$$

Проверяем, совпадают ли ответы на первых членах, сравнивая с изначальной рекуррентной формулой.
Примеры:
С кратностью корней

$f(n-2) = -6f(n-1) - 9f(n)$, где $f(0) = 1$, $f(1) = 5$
$r^2 = -6r - 9$
$r^2 + 6r + 9 = 0$
$(r+3)^2 = 0$, где степень 2 - и есть кратность
$r = -3$ (корень)
Общее решение:
$f(n) = n^0 \cdot c_1 \cdot (-3)^n + n^1 \cdot c_2 \cdot (-3)^n = (-3)^n \cdot (c_1 + n \cdot c_2)$
Степени у n соответсвуют кратности $[0, L-1] \cap \mathcal{Z}$
В данном случае $[0,1] \cap \mathcal{Z}$, то есть n^0 и n^1 у соответствующих корней
Напомним, что $f(0) = 1$, $f(1) = 5$

$$\begin{cases} (\text{-}3)^0 \cdot (c_1 + 0 \cdot c_2) = 1 \\ (\text{-}3)^1 \cdot (c_1 + 1 \cdot c_2) = 5 \end{cases}$$

Решив систему, получаем $c_1 = 1$ и $c_2 = -\frac{8}{3}$ Частное решение:
$f(n) = (-3)^n \cdot (1 - n \cdot \frac{8}{3})$
Проверка:
а) Рекуррент
$f(2) = -6 \cdot 5 - 9 = -39$
$f(3) = -6 \cdot (-39) - 9 \cdot 5 = 189$
$f(4) = -6 \cdot 189 - 9 \cdot (-39) = -783$
б) Явная формула
$f(2) = (-3)^2 \cdot (1 - 2 \cdot \frac{8}{3}) = -39$
$f(3) = (-3)^3 \cdot (1 - 3 \cdot \frac{8}{3}) = 189$
$f(4) = (-3)^4 \cdot (1 - 4 \cdot \frac{8}{3}) = -783$
Разные корни + как избавиться от свободного члена (просто число без $f(...)$)
$f(n+2) - 12f(n+1) + 27f(n) = 8$, где $f(0) = 4$, $f(1) = -2$
Чтобы избавиться от свободного члена (8) и всё равно получить верное решение, нужно увеличить каждое слагаемое в индексе на 1, затем вычесть из полученного выражения изначальное:

$$\begin{cases} \text{f(n+3) - 12f(n+2) + 27f(n+1) = 8} \\ \text{f(n+2) - 12f(n+1) + 27f(n) = 8} \end{cases}$$

Получаем: $f(n+3) - 13f(n+2) + 39f(n+1) - 27f(n) = 0$
$r^3 - 13r^2 + 39r - 27 = 0$
$(r-1)(r-3)(r-9) = 0$
$r_1 = 1$, $r_2 = 3$, $r_3 = 9$
Никакой кратности (одинаковых корней) у нас нет, значит у каждого слагаемого в общем решении будет множитель $n^0 = 1$

$f(n) = c_1 + c_2 \cdot 3^n + c_3 \cdot 9^n$

$$\begin{cases} \mathrm{f}(0) = \mathrm{c}_1 + c_2 + c_3 = 4 \\ \mathrm{f}(1) = \mathrm{c}_1 + 3c_2 + 9c_3 = -2 \\ \mathrm{f}(2) = \mathrm{c}_1 + 9c_2 + 81c_3 = -124 \end{cases}$$

Решаем систему (photomath тема) и получаем, что $c_1 = \frac{1}{2}$, $c_2 = \frac{17}{3}$, $c_3 = -\frac{13}{6}$

$f(n) = \frac{1}{2} + \frac{17}{3} \cdot 3^n - \frac{13}{6} \cdot 9^n$

Проверка:

а) Рекуррент

$f(2) = 8 + 12f(1) - 27f(0) = 8 + 12 \cdot (-2) - 27 \cdot 4 = -124$

$f(3) = 8 + 12f(2) - 27f(1) = 8 + 12 \cdot (-124) - 27 \cdot (-2) = -1426$

$f(4) = 8 + 12f(3) - 27f(2) = 8 + 12 \cdot (-1426) - 27 \cdot (-124) = -13756$

б) Явная формула

$f(2) = \frac{1}{2} + \frac{17}{3} \cdot 3^2 - \frac{13}{6} \cdot 9^2 = -124$

$f(3) = \frac{1}{2} + \frac{17}{3} \cdot 3^3 - \frac{13}{6} \cdot 9^3 = -1426$

$f(4) = \frac{1}{2} + \frac{17}{3} \cdot 3^4 - \frac{13}{6} \cdot 9^4 = -13756$

7. **Числа Фибоначчи. Определение, формула в замкнутом виде.**

Числа Фибоначчи - элементы числовой последовательности, в которой первые два числа равны 1 и 1, а каждое последующие число равно сумме двух предыдущих; однородное линейное рекуррентное соотношение второго порядка

$F_n = F_{n-1} + F_{n-2}$

Формаула в замкнутом виде:

$$F_n = \frac{\left(\dfrac{1+\sqrt{5}}{2}\right)^n - \left(\dfrac{1-\sqrt{5}}{2}\right)^n}{\sqrt{5}} = \frac{\varphi^n - (-\varphi)^{-n}}{\varphi - (-\varphi^{-1})} = \frac{\varphi^n - (-\varphi)^{-n}}{2\varphi - 1}$$

$\varphi = \dfrac{1+\sqrt{5}}{2}$ - золотое сечение

Формула без золотого сечения:

$$F_n = \frac{1}{2^n\sqrt{5}}\left((1+\sqrt{5})^n - (1-\sqrt{5})^n\right)$$

8. **Определение неоднородных линейных рекуррентных соотношений (НЛРУ). Общий алгоритм поиска решения.**

Опердeление:

Последовательность $u_1, u_2, ..., u_n, ...$ называется неоднородной линейной рекуррентной последовательностью порядка k, если существует натуральное число k, числа $a_1, a_2, ..., a_k$ (действительные или комплексные, причем $a_k \neq 0$) и функция $b_n = f(n)$ такие, что $u_{n+k} = a_1 u_{n+k-1} + a_2 u_{n+k-2} + ... + a_k u_n + b_n$ ($\leftarrow$ это и назы-

вается неоднородным линейным рекуррентным соотношением порядка k).
Алгоритм решения (примеры для лучшего понимая далее в билетах 10, 12, 13):
$g(n) = a_1 n^k + ... + a_{k+1}$
1) Решаем ОЛРУ
2) Рассмотрим частное решение нерекуррентной части
$f_p(n) = a_1 n^k + ... + a_{k+1} = g(n)$
Подставим данное $f_p(n)$ в исходное реккурентное уравнение
$f_a(f_p(n)) + f_{a-1}(f_p(n)) + ... + f_{a-k}(f_p(n)) = g(n)$
Соответствующие члены и будут членами частного решения нерекуррентной части. Если корень решения ОЛРУ равен 1, то при подстановке $f_p(n)$ в рекуррентное уравнение у нас оно может не сойтись $\Rightarrow$ $\mathrm{f}_p(n)$ (и будем так перебирать, пока не получится корректный ответ)
3) Общим решением такого НЛРУ будет общее решение ОЛРУ + частное решение нереккурентной части

10. **Поиск частного решения НЛРУ при функции-константе.**
См билеты 5-6, а именно пример *Разные корни + как избавиться от свободного члена (просто число без $f(...)$)*

12. **Поиск частного решения НЛРУ при функции-многочлене.**
Пример:
$f(n+2) - 2f(n+1) + f(n) = -9n - 1$
1) $r^2 - 2r + 1 = 0$
$(r-1)^2 = 0$
$r = 1$ (!)
$f_r(n) = c_1 + n \cdot c_2$ 2) $f_p(n) = an^2 + bn$
$(a(n+2)^2 + b(n+2)) - 2(a(n+1)^2 + b(n+1)) + (an^2 + bn) = -9n - 1$
$2a = -9n - 1$ - грусть печаль, нам такое не надо(
$f_p(n) = an^3 + bn^2$
$(a(n+2)^2 + b(n+2)^2) - 2(a(n+1)^3 + b(n+1)^2) + (an^3 + bn^2) = -9n - 1$
$6a_n + (6a + 2b) = -9n - 1$

$$\begin{cases} 6a = -9 \\ 6a + 2b = -1 \end{cases}$$

Получаем $a = -1,5$, $b = 4$
$f_p(n) = -1,5n^3 + 4n^2$
3) Общая формула:
$f(n) = f_r(n) + f_p(n) = c_1 + nc_2 - 1,5n^3 + 4n^2$
(Далее можно полчить частную формулу и проверить её, как мы это делаем в ОЛРУ)

Игорь (платница межбак)

13. **Поиск частного решения НЛРУ при функции-экспоненте.**
$g(n)$ - экспонента $= d \cdot (\alpha)^n$
1) Вновь отдельно решаем ОЛРУ, отбросив нереккурентную часть и получаем $f_r(n)$
2) В этот раз начинаем с $f_p(n) = d \cdot (\alpha)^n$ и, при необходимости, по аналогии с предыдущим алгоритмом перебираем варианты, умножением на n (обычно, если один из корней ОЛРУ равен α, то надо умножить на n b перебирать с $f_p(n) = d \cdot (\alpha)^n \cdot n$).
Так получаем частное решение нерекуррентной части для НЛРУ с экспонентой.
3) Вновь, $f(n) = f_r(n) + f_p(n)$
Пример:
$f(n+2) + 19f(n+1) + 90f(n) = 2 \cdot (-10)^{n+2}$
1) $r^2 + 19r + 90 = 0$
$(r+10)(r+9) = 0$
$r_1 = -10$, $r_2 = -9$
$f_r(n) = (-10)^n c_1 + (-9)^n c_2$
2) $f_p(n) = d \cdot (-10)^{n+2} \cdot n$
$(d \cdot (-10)^{n+4} \cdot (n+2)) + 19(d \cdot (-10)^{n+3} \cdot (n+1)) + 90(d \cdot (-10)^{n+2} \cdot n) = 2 \cdot (-10)^{n+2}$
Поделим на $(-10)^{n+2}$
$(d \cdot (-10)^2 \cdot (n+2)) + 19(d \cdot (-10) \cdot (n+1)) + 9(d \cdot n) = 2$
d = 0,2 $\Rightarrow$f$_p(n) = 0,2 \cdot (-10)^{n+2} \cdot n$
3) $f(n) = (-10)^n c_1 + (-9)^n c_2 + 0,2(-10)^{n+2} n$
(Далее можно полчить частную формулу и проверить её, как мы это делаем в ОЛРУ)

15. **Решение рекуррентных уравнений подстановкой на примере T(n) = aT(n/m) + bn, T(1) = b. «Угадывание» итогового решения.**

17. **Рекуррентные соотношения: основная теорема. Формулировка.**

18. **Оценка сумм через интеграл. Основная идея, оценки сверху и снизу.**

19. **Асимптотическая оценка** $\sum_{i=0}^{n-1} i^k$, $\sum_{i=0}^{n-1} \log i$, $\sum_{i=0}^{n-1} i \log i$.

21. **Алгоритм Евклида поиска НОД.**

22. **Расширенный алгоритм Евклида.**

23. **Понятие мультипликативного обратного. Поиск с помощью алгоритма Евклида.**

24. **Число бинарных деревьев с n вершинами в рекурсивной и нерекурсивной форме. Идея решения через производящие функции.**

25. **Понятие правильной скобочной последовательности (ПСП). Число ПСП длиной 2n.**

Эрнест

26. **Понятие пути Дика. Количество путей Дика длиной 2n.**

28. **Понятие сложности алгоритма. Лучший, худший, средний случай.**

29. **Ханойская башня. Алгоритм. Доказательство принадлежности к классу экспоненциальных за- дач.**

30. **Ханойская башня. Оценка сложности через решение рекуррентного уравнения.**

32. **Вычисление веса двоичного вектора. Полный перебор. Оценка сложности.**

33. **Вычисление веса двоичного вектора. Предвычисление. Оценка сложности.**

34. **Вычисление веса двоичного вектора со сложностью $\mathcal{O}(W(\mathrm{x}))$. Особенности реализации.**

36. **Задача коммивояжёра. Формулировка, вариации условий, матрица стоимости.**

37. **Формализация постановки задачи. Описание решения задачи коммивояжёра как перечисления гамильтоновых циклов.**

38. **Формализация постановки задачи. Описание решения задачи коммивояжёра как поиска га- мильтонова цикла.**

Амина

39. **Особенности асимптотической оценки сложности алгоритма.**

40. **Особенности точной оценки реализации алгоритма.**

41. **Определение методов частных целей, подъёма вверх, отрабатывания назад.**

42. **Применение методов разработки алгоритмов на примере задачи о джипе.**

43. **Отрабатывание назад. Задача о спичках.**

44. **Подъём вверх. Задача о миссионерах и каннибалах.**

45. **Частные цели. Задача о греческом кресте.**

46. **Подъём вверх. Задача о переливании.**

47. **Отрабатывание назад. Задача о пиратах.**

50. **Подъём вверх. Задача о лёгкой фальшивой монете. Обоснование поиска улучшенного решения.**

Гордей

51. **Задача о лёгкой фальшивой монете. Достаточное условие оптимальности решения.**

52. **Рекурсия. Задача о разбиении.**

54. **Рекурсия. Задача Иосифа Флавия. Рекуррентное решение.**

56. **Умножение однократной декомпозицией: идея, время работы.**

57. **Умножение рекурсивной декомпозицией: идея, время работы.**

58. **Связь умножения чисел и многочленов. Умножение за $\mathcal{O}(n^2)$.**

59. **Алгоритм Карацубы: идея, время работы.**

60. **Понятие кодов, сохраняющих разность.**

61. **Композиция. Построение кодов, сохраняющих разность, с квадратичным увеличением слов.**

62. **Композиция.Построениекодов,сохраняющихразность,сэкспоненциальным**

Остатки

63. **Эвристика. Метод ближайшего соседа в задаче коммивояжёра.**

64. **Эвристика. Задача о расписании процессоров.**