Алгоритмы и модели вычислений, группы 774-775, задание 1

- Кормен–Лейзерсон–Ривест–Штайн, главы 3 и 4 (параграфы 4.1, 4.2 и 4.6 можно пропустить). Оттуда нужно усвоить основы асимптотического анализа. Особое внимание обратите на основную теорему, мы впишем в семинарский тест задачи на неё.
- Кормен–Лейзерсон–Ривест–Штайн, пункт 9.3 (линейный в худшем случае алгоритм поиска порядковой статистики) и 2.3 (сортировка слиянием). На медиану тоже будут задачи в тесте.
- Любой источник в интернете по тому, как решать линейные рекуррентные соотношения. Например, этот: https://www.intuit.ru/studies/courses/65/65/lecture/1912?page=2. Не бойтесь математического английского! Он простой, серьёзно.

1. Оцените Θ-асимптотику рекурренты:

   (i) (пункт на применение основной теоремы)

   $$T(n) = 10T\left(\frac{n}{2}\right) + \frac{n^4}{\log n}.$$

   Если вам понадобится условие регулярности, вы обязаны чётко его доказать.

   (ii) (пункт, в котором основная теорема неприменима, теорема Акра-Баззи не является доказанным фактом в нашем курсе)

   $$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n\sqrt[3]{n}\log n).$$

   (iii) (пункт, в котором нужно вспомнить линейные рекурренты)

   $$T(n) = T(n-1) + 2T(n-2), \quad T(1) = T(2) = 1.$$

   Напоминание. В простейшем случае общее решение линейной однородной рекурренты вида $a_0T(n)+a_1T(n-1)+\ldots+a_pT(n-p) = 0$ ищут в виде $T(n) = c_1\lambda_1^n+\ldots+c_p\lambda_p^n$, где $\lambda_1, \ldots, \lambda_p$ – корни характеристического уравнения $a_0\lambda^p + a_1\lambda^{p-1} + \ldots + a_p = 0$. Однако в таком виде удаётся решить рекурренту только если среди корней $\lambda_1, \ldots, \lambda_p$ нет кратных. Если есть кратные корни, общая формула сложнее, она учитывает резонансные эффекты. Рекомендую вспомнить её из программы школы и/или первого курса, или же найти её в интернете (по ключевым словам “решение линейных рекуррентных соотношений”).

2. Рассмотрим детерминированный алгоритм поиска медианы по кальке известного линейного алгоритма, где используется разбиение массива на четвёрки элементов, в каждой из которых определяется *нижняя* медиана, т. е. из в каждой четверки выбирается второй по порядку элемент (элементы можно считать различными). Приведите рекуррентную оценку числа сравнений в этой процедуре и оцените сложность такой модификации.

3. Оцените трудоемкость рекурсивного алгоритма, разбивающего исходную задачу размера n на три задачи размеров $\lceil \frac{n}{\sqrt{3}} \rceil - 5$, используя для этого $10\frac{n^3}{\log n}$ операций.

4. Функция натурального аргумента $S(n)$ задана рекурсией:

   $$S(n) = \begin{cases} 100, & n \le 100 \\ S(n-1) + S(n-3), & n > 100 \end{cases}$$

   Оцените число рекурсивных вызовов процедуры $S(\cdot)$ при вычислении $S(10^{12})$.

5. Оцените трудоемкость рекурсивного алгоритма, разбивающего исходную задачу размера n на n задач размеров $\lceil \frac{n}{2} \rceil$ каждая, используя для этого $O(n)$ операций.

6. Дана рекурсивная программа (Данное деление в языке python3 целочисленно)

```
def f(x):
        if n > 1:
                print('Algorithm')
                print('Algorithm')
                print('Algorithm')
                f(n // 2)
                f(n // 4)
```

Пусть $g(n)$ обозначает число слов 'Algorithm', распечатанных в результате вызова $f(n)$

(i) Найдите в виде функции от n Θ-асимптотику $g(n)$.

(ii) Считая n степенью двойки, вычислите $g(n)$ точно.

7. Докажите, что для одновременного нахождения наибольшего и наименьшего среди n различных чисел необходимо не менее чем $3n/2 - 2$ сравнений. Начало решения задачи было разобрано на семинаре (Мы вводили некоторые 4 множества и функцию от них, а дальше смотели на изменение этой функции после выполнения операций сравнения)

Необязательные задачи

(можно сдавать до конца семестра)

1. Сравните по величине 2019-е члены рекуррентных последовательностей: $a_0 = b_0 = 0, \quad a_{n+1} = a_n^2 + 5, \quad b_{n+1} = b_n^2 + 2^n$ и найдите как можно более точно асимптотику последовательности a_n.