5 Демоническое программирование

5.1 Практика

1. **(!) Без двух подряд** Посчитать число последовательностей длины n из нулей и единиц, в которых не встречается две единицы подряд. $\mathcal{O}(n)$, в предположении что в модели вычислений арифметические действия работают за $\mathcal{O}(1)$.

2. Дан массив из n целых чисел и число d. Найти подпоследовательность максимальной длины с условием, что соседние элементы в ней должны отличаться не более чем на d. Решить за $\mathcal{O}(n^2)$.

3. Дан массив из n целых чисел, число d и число k. Найти подпоследовательность длины k с максимальной суммой элементов при условии, что соседние элементы в ней должны отличаться не более чем на d. Решить за $O(n^3)$.

4. **(!) НОП**
   Даны две последовательности длины n. Найдите наибольшую общую подпоследовательность этих последовательностей за $\mathcal{O}(n^2)$.

5. **(!) Рюкзаки**
   Даны n предметов. У каждого есть цена v_i и вес w_i. Найти max цену, которую можно набрать предметами суммарного веса $\leq S$. Время $\mathcal{O}(nS)$.

   (a) Каждый предмет можно брать сколько угодно раз. Память $\mathcal{O}(S)$.

   (b) Каждый предмет можно брать один раз. Память $\mathcal{O}(nS)$.

   (c) Каждый предмет можно брать один раз. Память $\mathcal{O}(S)$.

   (d) Каждый предмет можно брать один раз. Восстановить ответ.

6. **(!) НВП за $\mathcal{O}(n \log n)$**
   Найдите максимальную возрастающую подпоследовательность за $\mathcal{O}(n \log n)$. Найти длину и восстановить ответ.

7. **Задача про профессора**
   У профессора есть k яиц и n-этажное здание. Он хочет узнать такое максимальное x, что если яйцо бросить с x-го этажа, оно не разобьётся. Неразбившиеся яйца можно переиспользовать. Минимизировать число бросков в худшем случае.

   (a) $\text{poly}(n, k)$.

   (b) $\mathcal{O}(kn^2)$.

   (c) (*) $\mathcal{O}(kn \log n)$.

8. **(*) Разбиения на слагаемые**

   (a) Сколько способов разбить число n на k упорядоченных слагаемых?

   (b) Сколько способов разбить число n на k упорядоченных слагаемых, каждое не более c?

   (c) Сколько способов разбить число n на не более чем k неупорядоченных слагаемых?

   (d) Сколько способов разбить число n на k неупорядоченных слагаемых?

   (e) Сколько способов разбить число n на k неупорядоченных различных слагаемых?

9. **НОП revisited**
   Даны две последовательности длины n. Найдите наибольшую общую подпоследовательность этих последовательностей за $\mathcal{O}(n \log n)$, если известно, что в одной из последовательностей все элементы различны.

10. (*) Найти максимальное по весу паросочетание за $\mathcal{O}(n)$ на

    (a) дереве из n вершин,

(b) простом цикле из n вершин,

(c) связном неориентированном графе из n вершин и n рёбер.

Веса на рёбрах.

5.2 Домашнее задание

1. **(1) Лабиринт**
Дана матрица, в каждой клетке лежат монетки разной ценности. Некоторые клетки непроходимые. За один ход можно сместиться вверх или вправо. Рассмотрим все пути из левой нижней клетки в верхнюю правую.
    (a) **(0.5)** Найти число таких путей (можно считать ответ по модулю 10^9+7, чтобы в процессе вычислений не возникали длинные числа).
    (b) **(0.5)** Найти путь, сумма ценностей монет на котором максимальна/минимальна.

2. **$(1+0.5)$ И снова подпоследовательности**
Дан массив из n натуральных чисел: $a_1, \ldots, a_n$. Выберите подпоследовательность $i_1 \leq \ldots \leq i_k \in \{1, \ldots, n\}$ такую, что $l \leq |i_j - i_{j-1}| \leq r$ и $\sum_{j=1}^{k} a_{i_j} \to \max$.
    (a) **(1)** За $\mathcal{O}(n^2)$.
    (b) **(+0.5)** `(*)` За $\mathcal{O}(n)$.

3. **(1) Longest Common Prefix (LCP)**
Дана строка `s[0:n]` длины n.
Префикс строки s — это строка `s[0:i]` для какого-нибудь i (такой префикс называется i-м префиксом).
Суффикс строки s — это строка `s[i:n]` для какого-нибудь i (такой суффикс называется i-м суффиксом).

    Для каждой пары (i, j) найти длину наибольшего общего префикса i-го и j-го суффиксов строки s. $\mathcal{O}(n^2)$.

    Формально, $\mathrm{LCP}_{i,j} = \max\{k \mid 0 \leq k \leq n,\ i+k \leq n,\ j+k \leq n$ и `s[i:i+k] == s[j:j+k]`$\}$.

4. **(1)** Дан набор нечестных монеток с вероятностью выпадения орла $p_1, p_2, \ldots, p_n$. Требуется посчитать вероятность выпадения ровно k орлов за $\mathcal{O}(n \cdot k)$. Операции над числами считать выполнимыми за $\mathcal{O}(1)$.

5. **(1)** Пусть есть n подарков разной натуральной стоимости и три поросёнка. Нужно раздать подарки как можно честнее (так, чтобы минимизировать разность суммарной стоимости подарков самого везучего поросёнка и самого невезучего). Придумайте алгоритм решения данной задачи за $\mathcal{O}(nW^2)$, где W — суммарная стоимость подарков.

6. **(1)** На билете есть $2n$-значный номер. Билет считается счастливым, если сумма первых n цифр совпадает с суммой последних n цифр. По заданному числу n требуется найти количество счастливых $2n$-значных билетов за $\mathcal{O}(n^2)$. Считайте, что стандартные арифметические операции над числами выполняются за $\mathcal{O}(1)$.

    (Более формально, можно считать, что нужно вывести не ответ целиком, а ответ по модулю 10^9+7, чтобы в процессе вычислений не возникали длинные числа.)

7. **(+1.5)** `(*)` У профессора есть k яиц и n-этажное здание. Он хочет узнать такое максимальное x, что если яйцо бросить с x-го этажа, оно не разобьётся. Неразбившиеся яйца можно переиспользовать. Минимизировать число бросков в худшем случае.
    (a) **(+1)** $\mathcal{O}(kn^2)$.
    (b) **(+0.5)** $\mathcal{O}(kn\log n)$.