Практика по алгоритмам #9: динамика по подмножествам

1. Задачи

1. Операции с битами
   a) Дано w-битное число, за $\mathcal{O}(1)$, проверить, является ли число степенью двойки.
   b) Дано w-битное число, за $\mathcal{O}(1)$, проверить, правда ли, что в битовой записи никакие две единицы не идут подряд.
   c) Дано w-битное число, за $\mathcal{O}(\log w)$ найти старший единичный бит.
   d) Дано w-битное число, за $\mathcal{O}(\log w)$ посчитать количество единичных бит.
   e) Дано w-битное число, найти младший единичный бит быстрее чем $\mathcal{O}(w)$.
2. Задачи про паросочетания в произвольном графе
   a) Посчитать количество паросочетаний
   b) Найти максимальное по весу паросочетание
3. Для каждого множества вершин в графе посчитать количество независимых подмножеств.
   a) $\mathcal{O}(3^n)$
   b) $\mathcal{O}(2^n n)$
   c) $\mathcal{O}(2^n)$
4. Даны n купюр, у каждой есть ценность a_i, нужно распределить их среди k человек так, чтобы $\sum(s_i - A)^2 \to \min$. Здесь s_i — сколько получил i-й человек, $A = \frac{\sum a_i}{k}$.
5. Даны m подмножеств $\{1, 2, \ldots, n\}$, выбрать максимальное число непересекающихся за $\mathcal{O}^*(2^{\min(n,m)})$.
6. Есть k грузовиков, задача – перевести n вещей минимальным числом заездов. Один заезд – погрузить и отправить все грузовики.
   a) $k = 1$, $\mathcal{O}(3^n)$.
   b) $k = 2$, $\mathcal{O}(4^n)$.
   c) $\mathcal{O}(3^n k)$.
7. Посчитать для каждого $A \subseteq \{1, 2, \ldots, n\}$ $g[A]$ — сумму по всем $B \subseteq A$: $f[B]$.
   a) $\mathcal{O}(3^n)$
   b) $\mathcal{O}(2^n n)$
8. Покрыть строку s минимальным числом строк $t_i \in T$. Решение за $\mathcal{O}^*(2^{\min(|s|,|T|)})$.
9. Даны n точек на плоскости, покрыть их k кругами радиуса R, минимизировать R.
10. Найти количество подклик множества за
    a) $\mathcal{O}(2^n)$
    b) $\mathcal{O}(2^{n/2} n)$
    c) $\mathcal{O}(2^{n/2})$

2. Домашнее задание

2.1. Обязательная часть

1. (2) За $\mathcal{O}^*(2^n)$ проверить, можно ли покрасить граф в 4 цвета. Подсказка: в два цвета красить очень просто.
2. (2) Есть n вещей, у каждой есть стоимость v_i и вес w_i. Есть рюкзак, в котором можно у нести вещей суммарного веса не более W за один подход. За 4 подхода унести вещи максимальной суммарной стоимости. Время $O(3^n)$.
3. (2) Дан неориентированный граф. Посчитать за $\mathcal{O}^*(2^n)$ количество способов все вершины графа разбить на циклы. Каждая вершина должна лежать ровно в одном цикле. Разбиения на циклы различны, если различны множества использованных ребер.
4. (2) Дан двудольный граф, в первой доли $m \le 15$ вершин, во второй $n \le 1000$ вершин. Предложите алгоритм, который считает количество паросочетаний, покрывающих все m вершин первой доли (количество совершенных паросочетаний). Асимптотика заранее неизвестна, но таймлимит для реалиазации на языке `C++` – 1 секунда.
5. (3) Нужно составить расписание на один день в школе. Главная цель, чтобы все ушли домой пораньше, то есть минимизировать время окончания самого позднего урока. Уроки идут по расписанию, от звонка до звонка, все приходят утром к первому уроку. За день обязательно провести некоторое множество уроков Q, каждый урок q_i задается тройкой $\langle a, b, C\rangle$, где a — учебная группа, b — преподаватель, C — множество аудиторий, в которых можно провести занятие. В каждый момент времени преподаватель может вести не более чем одну группу, а группа, слушать не более чем одного преподавателя, также в одной аудитории нельзя одновременно проводить больше одного занятия. Составьте расписание.
   a) Время работы $\mathcal{O}(9^{|Q|})$ на 2 балла.
   b) Время работы $\mathcal{O}(3^{|Q|})$ на 3 балла.
   Для решения этой задачи полезно уметь за полином искать максимальное паросочетание в двудольном графе. Можно посмотреть формулировку задачи в википедии и сослаться на ее решение.
6. (3) У нас есть n типов товаров, суммарное число товаров A. Наш магазин занимается махинациями. В t-й момент времени, если у нас для каждого i от 1 до n есть хотя бы $a_{t,i}$ товаров типа i, мы можем взять $a_{t,1}$ товаров типа 1, $a_{t,2}$ товаров типа 2, $\ldots, a_{t,n}$ товаров типа n и одновременно заменить их на $b_{t,1}$ товаров типа 1, $b_{t,2}$ товаров типа 2, $\ldots, b_{t,n}$ товаров типа n. Причем махинации махинациями, а закон сохранения действует, поэтому $\forall t\colon \sum\limits_i a_{t,i} = \sum\limits_i b_{t,i}$. Махинацию t мы можем применить только один раз и только в t-й момент времени. Вопрос, можем ли мы после T моментов получить ровно c_1 товаров типа $1, \ldots, c_n$ товаров типа n (сумма c_i равна A). Решение за $\mathcal{O}(2^{A+n}T)$.

2.2. Дополнительная часть

1. (3) Дан граф $G = \langle V, E\rangle$ на n вершинах, найти для каждого $A \subseteq \{0, 1, \ldots, n-1\}$ $list[A]$ — множество всех максимальных по включению подклик A. Время работы должно быть $\mathcal{O}^*(M)$, где $M = \sum\limits_A |list[A]|$, т.е. оптимальным с точностью до полинома. $B \subseteq A$ называется подкликой, если $\forall i, j \in B\colon (i, j) \in E$. Подклика $B \subseteq A$ называется максимальной по включению, если $\forall k \in A \setminus B\colon A \cup \{k\}$ не является подкликой.
2. (4) Дана строки длины $n \le 10^6$ над алфавитом $k \le 26$. Нужно стереть все вхождения некоторых букв таким образом, чтобы не было $m \le n$ подряд идущих не стертых букв. Выберите минимальное по размеру подмножество k букв, которые нужно стереть. $\mathcal{O}(n+2^k)$.