АиСД Листок №3.
Алгоритмы решения трудных задач

Антюх Михаил группа 176

20 октября 2018 г.

Задача 1:

Пусть исходное множество S имеет размер n, разобьем наше множество на два подмножества по n/2 элементов в каждом. Для каждого подмножества построим все его возможные подмножества, их будет $2^{n/2}$ для каждого. Каждое их этих подмножеств представим в виде суммы их элементов. Получится два массива по $2^{n/2}$ элементов в каждом. Отсортируем оба массива за $2^{n/2} * n/2$. Имея два отсортированных массива мы можем проверить, может ли какой-то элемент из первого и второго массива в сумме дать t, для этого пойдем по первому массиву в порядке увеличения элементов, а по второму в порядке убывания, если текущий элемент из первого массива + текущий из второго $< t$, то переходим к следующему элементу в 1 массиве, если $> t$, то переходим к следующему элементу в 2 массиве, если $= t$, то алгоритм нашел решение и выдает true. Если мы дошли до конца обоих массивов и нужная пара чисел не найдена, то выдает false. Время проверки $2^{n/2}$. Время работы алгоритма: $O(2^{n/2} * n)$.

Задача 2: Запустим алгоритм A и B два раза, получим последовательность: x_1, x_2, x_3, x_4, где $x_i \in \{0, 1\}$. Вероятность того, что алгоритм A выдаст два нуля при условие, что в исходном графе есть клика равна 1/4, аналогично для алгоритма B. Наш алгоритм C должен выдавать 0 если алгоритм A выдал два нуля, а B выдал любую последовательность или алгоритм B выдал два нуля, а A выдал любую последовательность. Вероятность того, что наш алгоритм C выдаст 0 при условие, что в графе есть клика и независимое множество равна нужных размеров: $P(A \cup B) = P(A) + P(B) - P(A \cap B) = 1/4 + 1/4 - 1/16 = 7/16$ => наш алгоритм выдаст правильный ответ с вероятность 9/16 $> 1/2$

Задача 3: Если существует такой алгоритм, то существует дерево соответствующее этому алгоритму, в листьях дерева находятся ответы на задачу. Мы знаем, что алгоритм с вероятностью не менее 2/3 выдает правильный ответ, значит если в графе есть клика, то в дереве данного алгоритма будет как минимум 2/3 единиц. Алгоритм совершает не более $log_2 n$ рандомизированных шагов, выбирая между двумя вариантами перехода, все остальные действия детерминированные. Всего листьев в данном дереве $2^{log_2 n} = n$. Запустим алгоритм n раз и запретим ему идти по той ветви, которая уже была пройдена, в результате мы сможем посмотреть сколько у нас единиц в листьях и сказать точный ответ. Мы запускаем алгоритм n раз который работает за полином, в итоге получаем полином => мы умеем решать NP полную задачу за полином => мы можем решать все NP задачи за полином => P = NP

Задача 4: У нас на лекции был разобран рандомизированный и детерминированный алгоритм для вычисления означивания переменных, делающий не менее 7/8 истинных скобок в 3-КНФ. Пусть нам дана какая-та формула ф. Мы хотим чтобы $\frac{7}{8} \cdot p = 0.6$, где p какое-то кол-во скобок из формулы, p = $\frac{24}{35}$. Рассмотрим два случая, когда в формуле скобок размера ≤ 3 больше чем $\frac{24}{35}$ и когда таких скобок меньше или равно. В первом случае просто дополняем скобки размера < 3 так, чтобы они стали размера 3 и запускаем наш алгоритм на них, в итоге получаем, что $\frac{24}{35} \cdot \frac{7}{8} = 0.6$ скобок в среднем будут положительны. Во втором случае у нас скобок размера ≤ 3 меньше или равно чем 24/35, все остальные скобки размера больше чем 3 и в худшем случае все такие скобки при случайном означивание будут давать 1 с вероятность 15/16. Пусть всего скобок m, количество скобок размера $\leq 3 = \frac{24}{35} \cdot m$, остальных скобок $\frac{11}{35} \cdot m$, получаем в среднем $\frac{11}{35} \cdot m \cdot \frac{15}{16} + \frac{24}{35} \cdot m \cdot \frac{1}{2} = 0.635$m.

Задача 5: Нам дан набор ограничений, обозначим все ограничения как $l_1, l_2, \ldots, l_t$. Вероятность того, что ограничение l_i выполняется в случайной перестановке = $\frac{2}{3!} = \frac{1}{3}$. Пусть в оптимальном решение выполняется s ограничений и не выполняется t - s. В случайной перестановке у нас в среднем будет выполняться $\frac{s}{3}$ ограничений, что меньше в 3 раза чем в оптимальном решение. Значит для алгоритма который просто берет рандомную перестановку математическое ожидание числа ограничений, которым удовлетворяет вычисляемое им решение для заданных маркеров и ограничений, отличается от оптимального не более чем в 3 раза.