

Задача А

Условие

Найти значения переменных в формуле 3SAT, при которых наибольшее число скобок принимают истинное значение.

Доказательство NP

Допустим, что всего n переменных и m скобок.

1. Переформулируем задачу
Можно ли подобрать такие значения переменных, чтобы истинно было хотя бы K скобок?
2. Сертификат: последовательность $(k, x_1, x_2, \dots, x_n)$
3. Верификатор: Посчитать количество истинных скобок. Сравнить с k .

Доказательство NPH

Надо доказать, что $3SAT \leq_p MAX3SAT$. Допустим, что $MAX3SAT <_p 3SAT$, тогда у нас есть $(k, x_1, x_2, \dots, x_n)$ такое, что k скобок истинны, а значит хотя бы 1 истинна, то есть решена задача 3SAT, а значит допущение ($MAX3SAT <_p 3SAT$) неверное, то есть получено противоречие. Это значит, что исходное утверждение верно.

Задача G

Условие

Дана матрица A размера $M \times N$ из целых чисел, вектор B размера M из целых чисел. Найти бинарный вектор x , такой что $Ax \leq B$.

Доказательство NP

1. Сертификат: Бинарная последовательность чисел размера M . Очевидно, что размер сертификата равен M , то есть он полиномиален относительно размера самой задачи.
2. Верификатор: Непосредственно производит умножение и поэлементное сравнение. Самый очевидный алгоритм умножения матриц имеет сложность $O(n^3)$, $n = \max(M, N)$, сравнение выполняется за $O(M)$. А значит, верификатор работает за полиномиальное время от входа.

Доказательство NPH

Задача К

Условие

Дан взвешенный ориентированный граф $G = (V, E)$, найти в нем гамильтонов путь.

Доказательство NP

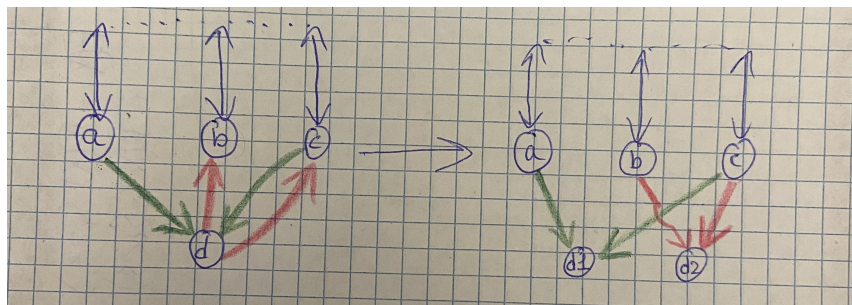
Допустим, всего N вершин.

1. Сертификат: Последовательность вершин размера N . Очевидно, что размер сертификата равен N , то есть он полиномиален относительно размера самой задачи.
2. Верификатор: Проверяет, посещены ли все вершины (достаточно проверить, что в сертификате все элементы уникальны) и существуют ли все ребра между соседними элементами в последовательности.

Доказательство NPH

Для доказательства этого факта сведем задачу о поиске гамильтонова цикла (далее - ГЦ) в ориентированном графе к задаче о поиске гамильтонова пути (далее - ГП) в ориентированном графе.

Пусть дан граф. Выберем произвольную вершину графа d и раздвоим ее на $d1$ и $d2$, так, что все входящие дуги в d теперь будут входить в $d2$, а все исходящие дуги из d теперь будут выходить из $d1$.



1. В одну сторону.

Утверждается, что если в исходном графе был ориентированный ГЦ, то в полученном будет ориентированный ГП. Докажем это утверждение. Допустим, что в исходном графе был ГЦ. Без ограничения общности будем считать, что он начинался и заканчивался в вершине d . Тогда после такого преобразования будет существовать ГП и он будет начинаться в $d2$ и заканчиваться в $d1$.

2. В обратную сторону.

Если в полученном графе будет ГП, то есть будет путь соединяющий все вершины, тогда он будет посещать вершины $d1$ и $d2$. Это значит, что он будет начинаться в $d2$ и заканчиваться в $d1$. Таким образом, если объединить эти вершины, то в полученном графе будет существовать ГЦ.

Использована информация из: [Ссылка на источник](#) .

Задача О

Условие

Дан набор задач A , для каждой есть время, за которое ее решает один процессор t_i , M процессоров и общий дедлайн D . Можно ли разбить все задачи на непересекающиеся множества задач A_j , такие что $\bigcup_{j=1}^n A_j = A$ и сумма t_i по каждому $A_j \leq D$. Формально: $\forall j \sum_{i \in A_j} t_i \leq D$

Доказательство NP

Допустим, что всего задач N .

1. Сертификат: последовательность $k_1, k_2, \dots, k_N : \forall i \ k_i \in \{1, 2, \dots, M\}$
2. Верификатор: Перебирает все процессоры, считает время работы на нем (суммирует те t_j , у которых k_j равно номеру этого процессора), если оно больше D хоть в одном случае, то возвращает ЛОЖЬ, иначе ИСТИНУ.

Доказательство NPH

Сведем задачу о рюкзаке к этой задаче. В задаче о рюкзаке у каждого объекта есть вес w_j и стоимость c_j . В нашей задаче $w_j = c_j = t_j$. Тогда

Задача R

Условие

Рассмотрим такой алгоритм построения максимальной клики: будем каждый раз удалять из графа вершину минимальной степени, пока не получим полный граф. Докажите или опровергните, что такой алгоритм дает решение, не более чем в X раз отличающееся от оптимального.

Решение

Утверждение. Если клика размера N , то все входящие в нее вершины имеют степень $= N - 1$. Доказательство очевидно.

Утверждение. Для любого n можно построить граф, в котором степень хотя бы одной вершины будет n , а всех остальных вершин будет $\geq n$, а размер максимальной клики будет 2.

Этот факт требует доказательства, но у меня не получилось это доказать (и найти доказательство тоже не получилось). Я построил для размера 2 и для размера 3. Кажется, что это утверждение истинно и доказать его можно.

Утверждение. Для любого N , существует граф, в котором есть клика размера N и используя этот алгоритм будет получаться клика размера 2. # Возьмем клику C размера $N + 1$. Возьмем граф G , в котором существует вершина v , степень которой N , а степень всех остальных вершин $\geq N$. Соединим C и G любым ребром, главное, чтобы оно не касалось вершины v . В получившемся графе минимальная степень вершины N и существует вершина степени N , входящая в граф C . Допустим алгоритм начнет удаление именно с этой вершины. Затем он будет постоянно удалять все вершины графа C , так как при удалении вершины графа C , степень всех вершин графа C уменьшится на 1. Таким образом граф C будет полностью удален и задача сведется к нахождению максимальной клики в графе G . А у этого графа максимальная клика имеет размер 2 по определению. #

Вывод

Последнее утверждение опровергает существование константы X .