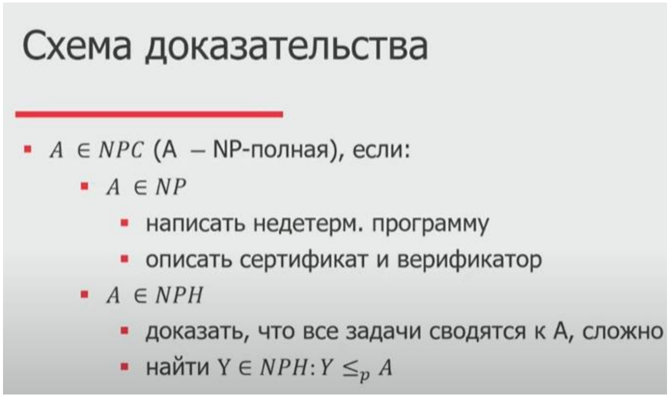
Вариант CDLPQV

Часть 1.

Для каждой задачи из этой части докажите ее принадлежность классу NP-complete. Можно набрать по пол балла за каждое из доказательств принадлежности классам NP и NP-hard.



C. Найти есть ли у формулы 3SAT решение, в котором в каждой скобке ровно одна переменная истинна.

Сертификат: переменные, использующиеся в задаче $x_1, x_2, \ldots, x_n$
Верификатор: вычисление функции $f(x_1, x_2, \ldots, x_n) = 1$

Назовем эту задачу 3SAT*, для доказательства ее принадлежности к NPH будем сводить ее к 3SAT.
В 3SAT все скобки должны быть равны 1, чтобы выражение было истинным. В нашей задаче в каждой скобке есть не более чем по одной 1, что не противоречит 3SAT. Таким образом 3SAT* — это частный случай 3SAT, но с обязательной 1 в каждой скобке. В то же время мы можем, не нарушая истинности формулы, заменять переменные истинными и наоборот пока истинна скобка, то есть пока в ней есть хотя бы одна переменная равная 1.

Пример перехода
$(x_1 \cup \overline{x_2} \cup x_3) \cap (x_2 \cup \overline{x}_1 \cup x_4)$
0 0 1 1 1 0
Заменим значения в переменных x_2, x_3
$(x_1 \cup \overline{x_2} \cup x_3) \cap (x_2 \cup \overline{x}_1 \cup x_4)$
0 1 0 0 1 0
В обратную сторону это тоже работает.

Сведение полиномиально, так как при переходе придется заменить не более чем n переменных.

D. Задача о рюкзаке. Есть n предметов, вес i-го предмета равен w_i. Нужно положить некоторые предметы из них в рюкзак так, чтобы суммарный вес взятых предметов был максимален, но не более S.

Сертификат: подмножество весов предметов
Верификатор: вычисление суммы масс подмножества предметов и сравнение результата с S

Задача аналогична разобранной на лекции (если правильно всё понял). Для принадлежности задачи к NPH закодируем веса предметов с помощью 3SAT. Построим множество из 2*(n+m) чисел, где n – число переменных, m – число скобок. Каждое число будет состоять из двух наборов разрядов размеров n и m. Для каждой переменной заведем пару чисел v_i и w_i , в которых 1 будет стоять в разряде номера переменной и в разрядах тех скобок, где переменная входит с отрицанием (без отрицания). Для каждой скобки b_i заведем два числа e_i и d_i с 1 в разряде этой скобке.

Пример:
$(x_1 \cup x_2 \cup \overline{x_3}) \cap (\overline{x}_1 \cup x_2 \cup \overline{x_3})$

	n			m	
	x1	x2	x3	b1	b2
v1	1	0	0	1	0
w1	1	0	0	0	1
v2	0	1	0	1	1
w2	0	1	0	0	0
v3	0	0	1	0	0
w3	0	0	1	1	1
e1	0	0	0	1	0
d1	0	0	0	1	0
e2	0	0	0	0	1
d2	0	0	0	0	1
S	1	1	1	3	3

Таким образом мы получили набор чисел, если мы можем с помощью них набрать нужную сумму S, значит формула удовлетворима. Раз 3SAT выражение имеет решение, то и данная модификация задачи о рюкзаке имеет решение.

Сведение полиномиально, так как из формулы мы получили 2*(n+m) чисел.

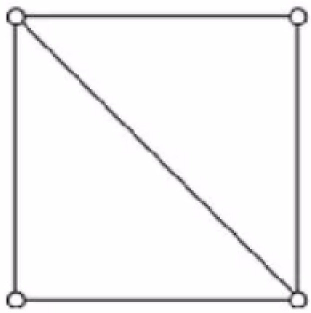
L. Дан взвешенный неориентированный граф $G = (V, E)$, найти в нем гамильтонов цикл.

Сертификат: сам цикл
Верификатор: проход по циклу с проверкой

Введем обозначения
HAM – граф с гамильтоновым циклом
WHAM – взвешенный неориентированный граф с гамильтоновым циклом

На лекции установили, что HAM принадлежит к NPC, поэтому будем сводить нашу задачу к HAM.

Сведём HAM к WHAM. Для примера возьмем этот граф:



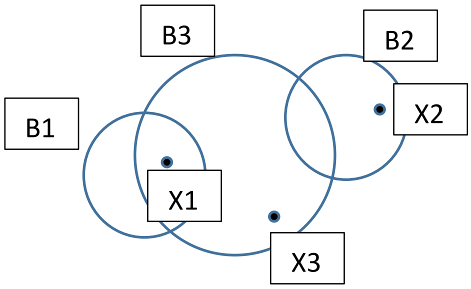
Имеющийся граф можно рассмотреть, как частный случай взвешенного графа, когда веса всех ребер равны 1, и это не будет противоречить условию задачи. Что касается ориентированности, то по умолчанию граф с гамильтоновым циклом неориентирован, поэтому об этом можно не говорить. Если бы нам пришлось сводить ориентированный граф к неориентированному графу, то тогда пришлось бы делать соответствующий переход.

Сведение полиномиально, оно проходит за число шагов, равное количеству ребер.

P. Дан набор точек P и k окружностей радиуса R. Можно ли расположить окружности на плоскости так, чтобы каждая точка из P лежала внутри хотя бы в одной окружности.

Сертификат: координаты точки и окружностей
Верификатор: проверка наличия точки на радиусах кругов

Сведем 3SAT к поставленной задаче, для примера возьмем следующий рисунок:



где x1, x2, x3 – точки, а в1, в2, в3 – круги.

Каждую точку мы принимаем за переменную в 3SAT, а каждый круг за скобку. Тогда в случае попадания точки в круг будем вносить ее в скобку с нормальным значением, а в случае непопадания с обратным.

$$(x_1 \cup \bar{x}_2 \cup \overline{x_3}) \cap (\bar{x}_1 \cup x_2 \cup \overline{x_3}) \cap (x_1 \cup \bar{x}_2 \cup x_3)$$

Также это можно представить в виде таблицы

	в1	в2	в3
x1	1	0	1
x2	0	1	0
x3	0	0	1

Несложно будет провести и обратную операцию — построить по формуле круги с точками.

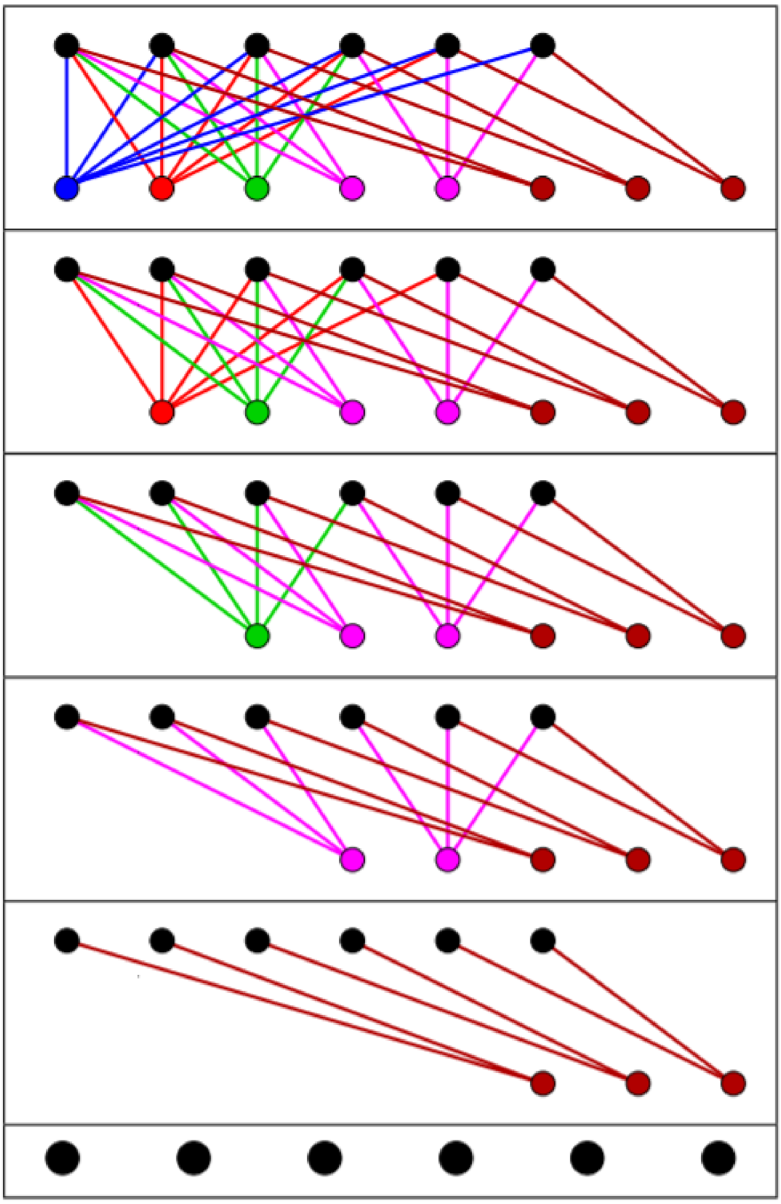
Сведение проходит за полиномиальное время, так как число переменных и скобок равно числу кругов и точек.

Часть 2.

Для задач этого раздела X является некоторой константой, требуется найти любой подходящий X и доказать, что алгоритм X-приближенный или что такой константы X не существует.

Q. Рассмотрим такой алгоритм построения минимального вершинного покрытия: будем каждый раз выбирать вершину, покрывающую максимальное количество еще не покрытых ребер. Докажите или опровергните, что такой алгоритм дает решение, не более чем в *X* раз отличающееся от оптимального.

Для доказательства рассмотрим один из наихудших случаев, когда наш алгоритм будет более всего отличаться от оптимального.



Рассмотрим следующий двудольный граф:
$G_n = (L \cup R, E)$, где L-множество из n вершин. Далее, для i = 2, . . . , n,
мы добавляем множество R_i из [n / i] вершин, к R, каждая из которых имеет степень i, и такие, что все они связаны между собой
с различными вершинами в R.

Алгоритм выполнения по шагам представлен на рисунке выше. На каждом шаге исключается вершина с наибольшим количеством непокрытых ребер.

Очевидно, что в G_n все вершины в L имеют степень по большей части n-1, так как
они связаны не более чем с одной вершиной R_i, для i = 2, . . . n. С другой стороны, существует вершина степени n в R (т. е. единственная вершина из R_n), таким образом, алгоритм сначала удалит эту вершину.

Мы утверждаем, что алгоритм удалит все вершины $R_2, \ldots, R_n$ и поместит их в вершинное покрытие. Чтобы увидеть это, обратим внимание, что если $R_2, \ldots, R_i$ еще активны, тогда все узлы R_i имеют степень i, все вершины L имеют степени максимум i-1, и все вершины $R_2, \ldots, R_{i-1}$ имеют степень строго меньше, чем i. Таким образом, алгоритм будет использовать вершины R_i. Все вершины R будут выбраны с помощью нашего алгоритма.

Оптимальное решение для графа на рисунке состоит в том, чтобы вобрать все вершины L в вершинное покрытие, что приводит к покрытию размера n. Но наш алгоритм выберет множество R.

$$|R| = \sum_{i=2}^{n} |R_i| = \sum_{i=2}^{n} \left\lfloor \frac{n}{i} \right\rfloor \geq \sum_{i=2}^{n} \left(\frac{n}{i} - 1 \right) \geq n \sum_{i=1}^{n} \frac{1}{i} - 2n = n(H_n - 2)$$

где $H_n = \sum_{i=1}^{n} 1/i = \lg n + \Theta(1)$

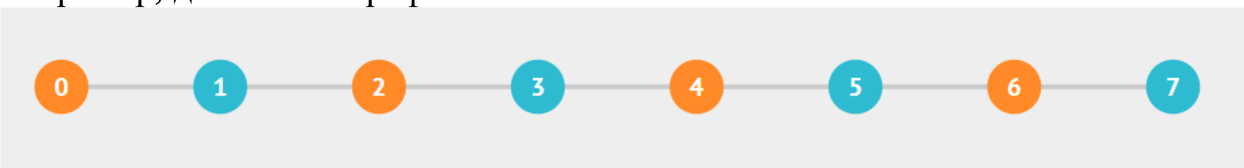
Таким образом искомое значение X будет равно

$$\frac{|R|}{|L|} = \frac{n(H_n - 2)}{n} = \ln n$$

Более чем в ln n раз решение заданного алгоритма не будет отличаться от оптимального.

Посчитаем разницу для примера с рисунка. Наш алгоритм убирал постепенно нижние вершины – 8 штук, оптимальный убрал бы верхние – 6 штук. 8/6 = 1.33, 1.33 < ln 6.

Могут быть и частные случаи, когда решение совпадет с оптимальным, например, для такого графа.



Часть 3.

Для задач этого раздела нужно предложить подходящий алгоритм, а потом доказать его оценку.

V. Предложите 2-приближенный алгоритм для решения следующей задачи. Есть n предметов, вес i-го предмета равен w_i. Нужно положить эти предметы в минимальное число коробок так, чтобы вес каждой коробки был не более S.