1 Постановка задачи.

Дан граф $G = (V, E)$. Необходимо перевести через реку $|V|$ предметов, не оставляя предметы соединенные ребром, без присмотра, имея лодку размера b. Формально, планом перевозки для лодки размера k называется последовательность $(L_i, R_i, B_i), i = 1..s$, такая что

- $L_i \sqcup R_i \sqcup B_i = V$
- L_i, R_i — независимые множества в G
- $L_1 \cup B_1 = V$
- $R_s \cup B_s = V$
- $L_k = L_{k-1}$ для четных $k \geq 2$
- $R_k = R_{k-1}$ для нечетных $k \geq 3$
- $|B_i| \leq b$.

Обозначим минимальное b для которого существует план перевозки $AN(G)$.

2 Структурная теорема

Пусть $VC(G)$ — размер наименьшего вершинного покрытия в G.

Легко проверить, что $VC(G) \leq AN(G) \leq VC(G) + 1$.

Мы будем использовать следующие факты, доказанные в статье (ссылка).

Теорема 1. *План для размера b существует тогда и только тогда, когда существует разбиение $V = X_1 \sqcup X_2 \sqcup X_3 \sqcup Y$, а также два непустых подмножества $Y_1, Y_2 \subset Y$, таких что*

- $X_1 \cup Y_1$, $X_2 \cup Y_2$, $X_1 \cup X_2 \cup X_3$ — *независимые*
- $|Y| \leq b$
- $|Y_1| + |Y_2| \geq |X_3|$

Теорема 2. *Если в графе есть хотя бы два минимальных вершинных покрытия, то $AN(G) = VC(G)$.*

3 Алгоритм для решения из статьи

- Найдем в графе G минимальное вершинное покрытие Y.
- Если $b < |Y|$, то вернуть «NO».
- Если $b \geq |Y| + 1$ то вернуть «YES».
- Если в графе G есть другое вершинное покрытие (не сложно проверить за b поисков вершинного покрытия), то вернуть «YES».
- Переберем $Y_1, Y_2 \subset Y$.
- В качестве X_1 возьмем вершины $V \setminus Y$ не связные с X_1, в качестве X_2 — вершины $V \setminus (Y \cup X_1)$ не связные с Y_2. В качестве X_3 все вершины $V \setminus (Y \cup X_1 \cup X_2)$.
- Проверим для полученных множеств утверждение теоремы. Если оно выполнено, вернуть «YES».
- Если для всех $Y_1, Y_2 \subset Y$ утверждение не выполнено вернуть «NO»

Итоговая сложность алгоритма $O^*(b \cdot T_{VC}(b) + 4^b)$, где T_{VC} — время на решение задачи о вершинном покрытии размера k.

4 Улучшенный алгоритм

Будем пытаться найти оптимальное Y_2 за время быстрее, чем 2^k.

После перебора Y_1 удалим из графа все вершины X_1. Обозначим новый граф $G' = (V', E')$, .

Посмотрим на множество $Z = X_3 \cup (Y \setminus Y_2)$. Оно является вершинным покрытием в графе G' как дополнение независимого $Y_2 \cup X_2$. Причем между множествами Z и тройками (Y_2, X_2, X_3) есть естественная биекция. Поэтому, можно вместо выбора Y_2 выбирать Z, при этом автоматически будут выполнены все условия, кроме условия на размеры Y_1, Y_2, X_3, и условия на непустоту Y_2. Перепишем условие на размер в терминах Z.

$$|Y_1| + |Y_2| - |X_3| = |Y_1| + |Y| - |Y \setminus Y_2| - |X_3| = b + |Y_1| - |Z| \geq 0$$

.

То есть задача выбора Z — это поиск вершинного покрытия в графе G' размера не более $b + |Y_1|$, отличного от Y'. Для решения этой задачи переберем вершину $v \in Y'$ которая не войдет в покрытие. Тогда мы обязаны взять всех ее соседей, после чего надо найти вершинное покрытие в графе $G \setminus (\{v\} \cup N_G(v) \cup X_1)$, размера меньше чем $b + |Y_1| - N_G(v)$. Заметим, что если $b + |Y_1| - N_G(v) \geq b - 1$, то такое покрытие точно найдется — $Z = Y \setminus \{v\}$, поэтому можно считать, что параметр задачи не превосходит b. Таким образом, задача решается за $O^*(b \cdot T_{VC}(b))$.

Суммарная сложность алгоритма $O^*(2^b \cdot b \cdot T_{VC}(b))$.

Известно, что вершинное покрытие можно решать за время $T_{VC}(b) = O^*(1.2738^b)$, в таком случае время работы алгоритма $O^*(2.5476^b \cdot b)$

5 Алгоритм за экспоненту от размера графа

5.1 За 4^n

Найдем вершинное покрытие Y в графе. Если оно не единственно, то все понятно.

Иначе, рассмотрим его дополнение X. Переберем все пары $X_1, X_2 \subset X, X_1 \cap X_2 = \varnothing$.

За Y_1 и Y_2 возьмем максимальное подмножества Y, такие, что $X_1 \sqcap Y_1, X_2 \sqcap Y_2$ независимы. Проверим для этого набора множеств условие теоремы.

5.2 За $4^{n-b} + 2^b$

За 2^b можно для всех подмножеств Y найти наибольшее независимое подмножество, с помощью динамического программирования.

Тогда вторую часть можно выполнять за $O^*(1)$ — надо взять самое большое независимое подмножество, среди эелементов Y, не соединенных ни с кем из X_1 или X_2.

5.3 за $2^{n-b} + 2^b$

Переберем все $X' \subset X$. Для каждого X', для которого есть непустое $Y_{X'} \subset Y$, такое что $X' \sqcup Y_{X'}$ независимо, выбрем максимальный $Y_{X'}$ по размеру и запишем моном $x^{bits(X')+2^{n-b}|Y|+2^{n-b+logb+1}|X'|}$, где $bits(X')$ — битовая маска длины $n-k$, соответствующая множеству.

Сложим все эти мономы. Мы получили многочлен степени $O^*(2^{n-b})$. Возведем его в квадрат за $O^*(2^{n-b})$ с помощью быстрого преобразования Фурье.

Рассмотим все мономы имеющие вид $x^{bits(X')+2^{n-b}t+2^{n-b+logb+1}|X'|}$. Они могли быть получены только как произведение двух мономов соответствующих непересекающимся X' (т.к размер сошелся). Среди всех таких необходимо выбрать моном с максимальным $t-(n-|X'|)$. Если эта величина больше 0, то $AC(G) = VC(G)$, иначе $AC(G) = VC(G)+1$. В качестве множеств X_1 и X_2 следует взять те, из которых был получен данный моном.

5.4 за $2^{n-b} \cdot VC^b$

Все тоже самое, только не предподсчитываем размеры, а решаем задачу для каждого множества отдельно.

5.5 Комбинируем!

Если $b \le 0.425n$, то воспользуемся алгоритмом за $2.5476^b \le 2.5472^{0.425n} \le 1.488^n$.

При $0.425n \le b \le 0.605n$ воспользуемся алгоритмом за $2^{n-b}+2^b$. $max(2^{n-b}, 2^b) \le 2^{0.605n} \le 1.52^n$.

При $b \ge 0.605n$ воспользуемся алгоритмом за $2^{n-b} * VC^b$ Оно убывает $2^{n-b} \cdot VC^b \le 2^{(1-0.605)n} \cdot 1.2738^{0.605n} = 1.521^n$.