Constructing a good C-path

Daniil Demidov, Maximilian Wittmer

17.11.2025

Abstract

We define the concept of a special subset of vertices of an undirected graph called a *clique* and set the rules for when one can“jump” from one clique to another. Such a pair is called a *move*. An (ordered) sequence of cliques, such that every pair of adjacent cliques form a move, is called a *good path* if it satisfies specific properties that we define later. The goal is to establish when a good path is exists and, if so, how to construct it.

1 Introduction

Since we work with graphs, we need to outline a convention we will use throughout this document.

Convention 1.1: Let $\Gamma = (V, E)$ be an simple finite undirected graph with V being the set of vertices, and $E \subset \mathcal{P}(V)$ being the set of edges $e = \{x, y\} \in E$ for some $x, y \in V$.

Definition 1.2: *(Link)* For some $v \in V$, we call $lk_\Gamma(v) := \{v\} \cup \{u \in V \mid \{v, u\} \in E\}$ the **link** of v.

1.1 Joins

We need to define a specific type of graph that will be studied later.

Definition 1.1.1: *(Join Decomposition)* We call a tuple (L, O) with $L, O \subseteq V$ a **join decomposition**, if the following conditions are satisfied:
(i) $V = L \sqcup O$
(ii) $\forall l \in L, \forall o \in O : \{l, o\} \in E$.

Definition 1.1.2: *(Join Graph)* If, for a graph Γ, there exists a join decomposition, then we call Γ a **join**.

1.2 Cliques

Now, we define the already mentioned concept of special subsets of V.

Definition 1.2.1: *(Clique)* A subset $\alpha \subset V$ is called a **clique** if $\forall x, y \in \alpha, \{x, y\} \in E$.

Notation 1.2.2: *(Set of All Cliques)* For a graph Γ, we denote the set of all cliques as $\mathcal{C}(\Gamma) := \{\alpha \in \mathcal{P}(V) \mid \forall x, y \in \alpha, \{x, y\} \in E\}$.

Definition 1.2.3: *(Move)* Let $\alpha, \beta \in \mathcal{C}(\Gamma)$. We write $\alpha \to \beta$ if and only if

$$(\beta \setminus \alpha) \cap \left(\bigcap_{v \in \alpha} lk_\Gamma(v) \right) = \emptyset \tag{1}$$

and say, that we can *move* from α to β.

Remark 1.2.4: For all $\alpha, \beta \in \mathcal{C}(\Gamma)$:
(i) $\beta \subseteq \alpha \implies \alpha \to \beta$;
(ii) Specifically, $\alpha \to \alpha$

1.3 Paths

Definition 1.3.1: *(C-path)* A sequence of cliques $(\omega_n)_{n\in\mathbb{N}}$ is called a **c-path** (**c**liques **path**), if $\omega_{n+1} \to \omega_n$ holds for all $n \in \mathbb{N}$.

At first, it might seem counterintuitive to define a c-path in a reverse order. But it is motivated by the goal of finding an algorithm for constructing such a path, which will be done top-to-bottom.

Remark 1.3.2: The definition of a c-path requires it to be an *infinite* sequence of cliques which would not translate to any real world application. So, at first glance, it seems impossible to have a finite path of cliques expressed as a c-path. But this problem can easily be solved by setting $\omega_n := \alpha$ for all $n \in \mathbb{N}, n \geq m$, whereby $m \in \mathbb{N}$ is the length of the path and α is the starting clique. Indeed, it satisfies the definition of a c-path because of Remark 1.2.4.

Definition 1.3.3: *(A good c-path)* A c-path ω is called **good** iff
(i) ω_1 is maximal
(ii) $\exists M \in \mathbb{N} : |\omega_M| = 1$.

Definition 1.3.4: *(A good vertex)* $v \in V$ is called a **good vertex** if there exists a good c-path ω, such that $\omega_M = \{v\}$ for some $M \in \mathbb{N}$.

1.4 Goal

The goal is to prove the following theorem:

Theorem 1.4.1: A graph Γ is a join $\iff$ no vertex of Γ is good.

Proof:
- "$\implies$" has been shown in our previous paper. Write it down anyway?
- "$\impliedby$" Is equivalent to Lemma 1.4.2.

■

Lemma 1.4.2: A graph Γ is not a join $\implies$ there is a good vertex.

The rest of this paper is dedicated to proving Lemma 1.4.2. We will do is constructively by describing an algorithm for finding a c-path to an arbitrary maximal clique and proving that it is good.

2 Algorithms

2.1 Finding a Join Decomposition

The first goal is to determine whether a given graph Γ is a join or not.

Construction 2.1.1: *(Finding a Join Decomposition)* The goal is to construct two sequences $L_n, O_n \in \mathcal{P}(V)$, such that $\forall n \in \mathbb{N}_0$

$$L_n \cap O_n = \emptyset \tag{2}$$

$$L_n \cup O_n = V \tag{3}$$

We start by choosing an arbitrary vertex $v \in V$. We set $L_0 := \{v\}$, $O_0 := V \setminus \{v\}$. Thus, both (2) and (3) are satisfied. For $n \in \mathbb{N}$, we define

$$L_n = L_{n-1} \cup \{o \in O_{n-1} \mid \exists l \in L_{n-1}, \{l, o\} \notin E\} \tag{4}$$

$$O_n = V \setminus L_n. \tag{5}$$

Note that (5) implies (2). Inductively, (3) also holds.

However, the algorithm must terminate. To determine when the recursion has to be stopped, we introduce $\Delta L_n := L_{n+1} \setminus L_n$ for all $n \in \mathbb{N}_0$. If $\Delta L_s = \emptyset$ for some $s \in \mathbb{N}_0$, then we have $L_{s+1} = L_s$. That means that *either* $O_s = \emptyset$ *or* $O_s \neq \emptyset$ (then, (4) would imply that $\forall o \in O_s, \forall l \in L_s : \{l, o\} \in E$). Both cases imply that $\forall n \geq s : \Delta L_n = \emptyset$. That is why it makes sense to stop the recursion as soon as $\Delta L_n = \emptyset$. The recursion depth is represented by $J_\Gamma(v) := \min\{s \in \mathbb{N}_0 \mid \Delta L_n = \emptyset \text{ wenn } L_0 = \{v\}\}$. Proposition 2.1.2 shows that $J_\Gamma(v)$ always exists.

Proposition 2.1.2: $\forall v \in V, \exists s \in \mathbb{N} : s = J_\Gamma(v)$.

Proof: By construction, s exists $\iff \Delta L_n = \emptyset$ for some $n \in \mathbb{N}$. Suppose that, for some starting vertex $v \in \Gamma$, $\Delta L_n \neq \emptyset$ for all $n \in \mathbb{N}$. That would imply $|L_{n+1}| > |L_n|$ for all $n \in \mathbb{N}$. This contradicts $|V| < \infty$ since $L_n \subseteq V$ for all $n \in \mathbb{N}$. Thus, the assumption that such vertex $v \in V$ exists, that $J_\Gamma(v) \notin \mathbb{N}$, is false. ■

Now, the goal is to show that (L_s, O_s) is indeed a join decomposition of Γ.

Proposition 2.1.3: Γ is a join $\iff$ the algorithm described in Construction 2.1.1 terminates with $O_s \neq \emptyset$ for all $v \in V$, whereby $s = J_\Gamma(v)$.

Proof: "$\Longleftarrow$": Assume that $\forall v \in V, O_s \neq \emptyset,$ whereby $s = J_\Gamma(v)$. According to Construction 2.1.1, we have $\emptyset \neq L_s \subsetneq V$, such that $L_s \cap O_s = \emptyset$ and $\forall o \in O_s, \forall l \in L_s : \{l, o\} \in E$. By definition, Γ is a join.

"$\Longrightarrow$": Let Γ be a join graph. There could be different join decompositions (L, O) of this graph. As a reminder, a join decomposition of a graph is a tuple $(L, O) \in \mathcal{P}(V)^2$, such that

$$L \cap O = \emptyset \ , \ \ L \cup O = V \text{ and } \forall o \in O, \forall l \in L : \{l, o\} \in E. \tag{6}$$

Now, let us choose an arbitrary vertex $v \in V$. Then, there exists a join decomposition $(L, O) \in \mathcal{P}(V)^2$ such that $v \in L$. Join decomposition is not unique, so we choose (wlog) such a decomposition that $|L|$ is minimal across all possible decomposition. Now we let the algorithm Construction 2.1.1 run with $L_0 = \{v\}$. By induction we show that $O \subseteq O_n$ and $L_n \subseteq L$ for all $n \in \mathbb{N}_0$.

Clearly, we have $O \subseteq O_0$, as well as $L_0 \subseteq L$. Now we assume that $O \subseteq O_n$ and $L_n \subseteq L$ for some $n \in \mathbb{N}$. By construction, $\forall o \in O_{n+1}, \forall l \in L_n : \{l, o\} \in E$, therefore the assumption $L_n \subseteq L$ implies $O \subseteq O_{n+1}$. By construction, $L_{n+1} \cap O_{n+1} = \emptyset$. By definition, $L \cap O = \emptyset$. Thus, $\forall l \in L_{n+1} : l \notin O_{n+1} \Longrightarrow l \notin O \Longleftrightarrow l \in L$, which means $L_{n+1} \subseteq L$.

Obviously, $\forall n < s = J_\Gamma(v), \Delta L_{n+1} \neq \emptyset \Longrightarrow |L_n| < |L_{n+1}| \leq |L|$ as well as $|O| \geq |O_n| > |O_{n+1}|$. Proposition 2.1.2 shows, that the algorithm stops after $s = J_\Gamma(v))$ steps. Hence, $|L_s| \leq |L| < |V| \Longrightarrow L_s \neq V \Longrightarrow O_s \neq \emptyset$. That means that $\forall l \in L_s, \forall o \in O_s, \{l, o\} \in E \Longrightarrow L_s = L$ as well as $O_s = O$. ■

2.2 Adjacency Table

Now, we introduce one last construction that will be used to prove Lemma 1.4.2.

Construction 2.2.1: *(Adjacency Table)* Let $T_\Gamma(v)$ be a tuple of $(s+1)$ subsets of V, whereby $s = J_\Gamma(v)$ and

$$(T_\Gamma(v))_i := \Delta L_{i-1} = L_i \setminus L_{i-1} \quad \text{for} \quad i \in \{1, ..., s\} \quad \text{and} \quad (T_\Gamma(v))_0 := \{v\} \tag{7}$$

.

We call $T_\Gamma(v)$ the **adjacency table** of v (over Γ).

Example 2.2.2: Let $\Gamma = (\{1, 2, ..., 10\}, E)$. Then, the adjacency table might look something like

$(T_\Gamma(1))_0$	$(T_\Gamma(1))_1$	$(T_\Gamma(1))_2$	$(T_\Gamma(1))_3$	$(T_\Gamma(1))_4$
1	3 7	4 2 8	6 9	5 10

Convention 2.2.3: From now on, let $\Gamma = (V, E)$ be a non-join graph, q an arbitrary, but fixed vertex of Γ (as in Construction 2.1.1). Let $s = J_\Gamma(q) \in \mathbb{N}$ be the number of steps which the algorithm (Construction 2.1.1) terminates after with $L_s = V, O_s = \emptyset$. Let $T := T_\Gamma(q)$ be the adjacency table of this vertex.

Let $I := \{1, ..., s\}$ and $I_0 = I \cup \{0\}$ be the index sets for the adjacency table T.

Now, let's make the following observation:

Lemma 2.2.4: $\forall i, j \in I_0 : i \neq j \Rightarrow T_i \cap T_j = \emptyset$.

Proof: Without loss of generality, we assume $i > j$. By construction, we have then $T_j \subseteq L_{i-1}$ and $T_i = \Delta L_{i-1} = \{o \in O_{i-1} \mid \exists l \in L_{i-1} : \{l, o\} \notin E\} \subseteq O_{i-1}$. Because of $L_{i-1} \cap O_{i-1} = \emptyset$ we obtain $T_i \cap T_j = \emptyset$. ■

Lemma 2.2.5: For each adjacency table, the following holds:

$$\forall i \in I : \forall w \in T_i \; \exists v \in T_{i-1} : \{v, w\} \notin E \tag{8}$$

and

$$\forall i, j \in I_0 : |i - j| > 1 \Longrightarrow \forall v \in T_i, w \in T_j : \{v, w\} \in E. \tag{9}$$

Proof: (8) follows directly from Construction 2.1.1 and Construction 2.2.1. ■

Proof: (9): We assume that $\exists i, j \in I_0$ with $|i - j| > 1$, such that $\exists v \in T_i, w \in T_j : \{v, w\} \notin E$. Without loss of generality, let $i > j$. Lemma 2.2.4 implies $v \neq w$ and $T_i \cap T_j = \emptyset$ as well as $T_j \cap T_{j+1} = \emptyset$. $w \in T_j = \Delta L_{j-1} \Rightarrow$ by Construction 2.1.1, $w \in L_j$. We know that $T_{j+1} = \Delta L_j = \left\{o \in O_j \mid \exists l \in L_j : \{l, o\} \notin E\right\}$. $v \in T_i$ and $i > j$ imply $\forall k \leq j < i, v \notin T_k$, which means that $v \in O_j \Longrightarrow v \in \Delta L_j = T_{j+1} \Longrightarrow T_i =$

$T_{j+1} \implies j+1 = i$. That contradicts $|i-j| > 1$, thus, the assumption that such v and w exist is false. That completes the proof of (9). ■

Now our goal is to take advantage of the adjacency table defined above and its properties. For readability, we call the entries of the adjacency table T_i *cells*. The direction "left to right" corresponds to the direction "0 to s".

2.3 Constructing a Good C-Path in $\mathcal{O}(|V|)$

Notation 2.3.1: *(Selection)* We set $\Omega_n^i = \omega_n \cap T_i$ and call Ω_n^i the **selection** from the cell T_i in step n.

We say that a vertex v is **selected** in step n if $v \in \omega_n$.

From now on, let μ be an arbitrary (possibly maximal) clique of Γ. The goal is to construct a c-path $\alpha = \omega_1 \leftarrow \omega_2 \leftarrow \dots \leftarrow \omega_m = \{q\}$.

Construction 2.3.2: Set $\omega_1 := \alpha$. Then we proceed as follows to construct ω_{n+1} if ω_n has already been constructed:

1. We denote the set of indices of all even or odd (depending on the oddness of n) problematic cells (which should not contain any selected vertices in the end) as
$$p_n := \left\{ i \in I \mid \Omega_n^i \neq \emptyset \wedge i \equiv n \bmod 2 \right\} \tag{10}$$
2. Then, we choose exactly one vertex $r_j \in \Omega_n^j$ $(j \in p_n)$ that shall not be in ω_{n+1}.
3. By Lemma 2.2.5, for every $j \in p_n$ there exists at least one vertex $x_j \in T_{j-1}$ such that $\left\{x_j, r_j\right\} \notin E$. Let x_j be such a vertex for all $j \in p_n$. Set $\chi_n := \left\{ x_j \mid j \in p_n \right\}$. Note that $|\chi_n| = |p_n|$ since the cells are disjoint (Lemma 2.2.4).
4. Define ω_{n+1} as follows:
$$\omega_{n+1} := \chi_n \cup \left(\omega_n \cap \left(\bigcap_{x \in \chi_n} lk_\Gamma(x) \right) \right). \tag{11}$$
By construction, $\forall j \in p_n, r_j \notin \omega_{n+1}$ as we wished. Now we show that ω is a valid c-path.

Proposition 2.3.3: ω from Construction 2.3.2 is a valid **good** c-path, namely, $\forall n \in \mathbb{N}$:

(i) $\omega_n \in \mathcal{C}(\Gamma)$,

(ii) $\omega_{n+1} \to \omega_n$,

(iii) $|\omega_{n+1}| \leq \omega_n$ and $\exists k_n \in \mathbb{N} : |\omega_{n+k_n}| < \omega_n$ which means that $\exists m \in \mathbb{N} : |\omega_m| = 1$.

Specifically, $m \leq 2(|V|+1)$.

Proof:

(i) We show that by induction. If $n = 1$, there is nothing to show. Assume that for some $n > 1, \omega_n \in \mathcal{C}(\Gamma)$. Then we have three cases for an arbitrary pair of vertices $x, y \in \omega_{n+1}$:
- $x, y \in \chi_n$. Because only even or odd cells were considered by choosing p_n in each step n, there exist $i, j \in I : |i-j| > 1$ such that $x \in T_i$ and $y \in T_j$. Lemma 2.2.4 implies that $\{x, y\} \in E$.
- $x \in \chi_n, y \in \left(\omega_n \cap \left(\bigcap_{x \in \chi} lk_\Gamma(x)\right)\right)$. By definition of the link, $y \in lk_\Gamma(x)$ implying $\{x, y\} \in E$.
- $x, y \in \left(\omega_n \cap \left(\bigcap_{x \in \chi} lk_\Gamma(x)\right)\right) \subset \omega_n \in \mathcal{C}(\Gamma)$. The definition of a clique implies that $\{x, y\} \in E$.

(ii) Clearly, we have

$$\omega_n \setminus \omega_{n+1} = \omega_n \cap \left(\omega_n \cap \left(\bigcap_{x\in\chi} lk_\Gamma(x)\right)\right)^C = \omega_n \cap \chi_n^C \cap \left(\bigcap_{x\in\chi} lk_\Gamma(x)\right)^C. \tag{12}$$

That and $\bigcap_{v\in\omega_{n+1}} lk_\Gamma(v) \subset \bigcap_{v\in\chi_n} lk_\Gamma(v)$ means that

$$\begin{aligned}(\omega_n \setminus \omega_{n+1}) \cap \left(\bigcap_{v\in\omega_{n+1}} lk_\Gamma(v)\right) = \\ = \omega_n \cap \chi_n^C \cap \left(\bigcap_{x\in\chi} lk_\Gamma(x)\right)^C \cap \left(\bigcap_{v\in\omega_{n+1}} lk_\Gamma(v)\right) = \emptyset \Longleftrightarrow \omega_{n+1} \to \omega_n.\end{aligned} \tag{13}$$

(iii) Because of $|\chi_n| = |p_n|$ and $|\omega_n \cap \left(\bigcap_{x\in\chi} lk_\Gamma(x)\right)| \leq |\omega_n| - |p_n|$, we conclude $|\omega_{n+1}| \leq |\omega_n|$. In every other step, one vertex from the right-most selection is being removed and, at most, one element is added to the selection of the cell to the left of it. That means, that each element "traces a path" to the left of the table. Because the table is finite, the right-most selection becomes empty in no more than $2 \cdot |\Omega_n^i|$ steps, where i is the index of this selection. Then, there is a new "right-most" selection with an index *strictly smaller* than i. The same argument applies to this new selection. Thus, after $m \leq 2 \cdot (|V| + 1)$ steps, $\omega_m = \{q\}$.

■

Corollary 2.3.4: The algorithm from Construction 2.3.2 yields a good c-path of the length $\mathcal{O}(|V|)$ for non-join graphs.

2.4 The Optimal Algorithm

Now, we will answer the question: is there a better algorithm in terms of asymptotic path length with respect to $|V|$? Spoiler: no. Here is the theorem we will prove in this section:

Theorem 2.4.1: The worst-case length of a c-path generated by any algorithm lies between $\frac{|V|-1}{2}$ and $2(|V|+1)$, thus, being $\mathcal{O}(|V|)$.

We are going to prove Theorem 2.4.1 by contradiction. For that, we would need some new tools.

From now on, let all graphs $\Gamma = (V, E)$ have the set of vertices being a subset of consecutive natural numbers starting from 1, namely $V = \{1, ..., |V|\}$. Clearly, every finite graph is equivalent to a graph of such form. Then, the following definition makes sense:

Definition 2.4.2: For a clique $\alpha \in \mathcal{C}(\Gamma)$, we call $\varnothing(\alpha) := \max\alpha - \min\alpha$ the *diameter* of α.

Example: $\varnothing(\{1,3,5\}) = 5 - 1 = 4$.

Construction 2.4.3: We want to construct a sequence of graphs $(\Gamma_n)_n$ recursively defining the sets of vertices $(V_n)_n$ and edges $(E_n)_n$, and setting $\Gamma_n := (V_n, E_n)$ for $n \in \mathbb{N}$.

(i) Set $V_1 = \{1\}$ and $E_1 = \emptyset$;

(ii) Then, define recursively

$$\begin{aligned} V_{n+1} &:= V_n \cup \{n+1\} = \{1, ..., n+1\}, \\ E_{n+1} &:= E_n \cup \{\{n+1, v\} \mid v \in V_n \setminus \{n\}\}. \end{aligned} \tag{14}$$

Remark 2.4.4: If we run the join decomposition algorithm (Construction 2.1.1) on the vertex 1 for each Γ_n, we obtain an adjacency table of the following form:

1	2	3	...	n

Then, if we illustrate a clique α by highlighting its vertices with an underline, the diameter of alpha is simply the “distance” in the table between the right-most and the left-most highlighted vertices. E.g. considering $\alpha = \{3, 5, 7\}$, we clearly see that $\varnothing(\alpha) = 4$:

...	**<u>3</u>**	4	**<u>5</u>**	6	**<u>7</u>**	...

Now, we would like to discuss some of the properties of arbitrary clique paths in Γ_n.

Lemma 2.4.5: Let Γ_n be some graph as in Construction 2.4.3 for some $n \in \mathbb{N}, n \geq 5$ and $\left(\alpha_j\right)_j$ an arbitrary sequence of cliques, such that $|\alpha_1| = 1$ and $\alpha_j \to \alpha_{j+1}$ for all $j \in \mathbb{N}$. Then, $\varnothing\left(\alpha_j\right) \leq 2j$ for each $j \in \mathbb{N}$.

Proof: We show the statement by induction. By construction, we have $\varnothing(\alpha_1) = 0 \leq 2$. Now we assume that $\varnothing\left(\alpha_j\right) \leq 2j$ holds for some $j \in \mathbb{N}$. We required $\alpha_j \to \alpha_{j+1}$ which is equivalent to $\left(\alpha_{j+1} \setminus \alpha_j\right) \cap \bigcap_{v \in \alpha_j} lk_\Gamma(\mathtt{v}) = \emptyset$. That implies

$$u \in \alpha_{j+1} \Longrightarrow u \in \alpha_j \text{ or } u \notin \bigcap_{v \in \alpha_j} lk_\Gamma(\mathtt{v}) \tag{15}$$

(15) is equivalent to

$$u \in \alpha_{j+1} \Longrightarrow u \in \alpha_j \text{ or } u \in \bigcup_{v \in \alpha_j} [lk_\Gamma(\mathtt{v})]^C \tag{16}$$

Since $lk_\Gamma(\mathtt{v}) = \{v-1, v+1\}^C$ by construction, we conclude that (15) is equivalent to

$$u \in \alpha_{j+1} \Longrightarrow u \in \bigcup_{v \in \alpha_j} \{v-1, v+1\} \cap V \tag{17}$$

That implies $\min \alpha_{j+1} \geq \min \alpha_j - 1$ as well as $\max \alpha_{j+1} \leq \max \alpha_j + 1$, which means that $\varnothing\left(\alpha_{j+1}\right) \leq \max \alpha_j - \min \alpha_j + 2$, and, using our assumption, we conclude that $\varnothing\left(\alpha_{j+1}\right) \leq 2j + 2 = 2(j+1)$.

■

Lemma 2.4.6: Every maximal clique of Γ_n has diameter not smaller than $n - 1$.

Proof: Assume there is some $n \in \mathbb{N}$ such that there exists a maximal clique α with $\varnothing(\alpha) < n - 1$. That means that $v := \max(\alpha) \leq n - 2$. By construction, $n \in V_n$ and $\{k, n\} \in E$ for all $k \in \alpha$. That implies that $\alpha \cup \{n\}$ is a clique. That contradicts the assumption that α is maximal. ■

Proposition 2.4.7: For every sequence of cliques $\alpha_1 \to ... \to \alpha_m$, such that $|\alpha_1| = 1$ and α_m is maximal, $m \geq \frac{n-1}{2}$.

Proof: Lemma 2.4.6 implies that $\varnothing(\alpha_m) \geq n-1$. By Lemma 2.4.5, $2m \geq \varnothing(\alpha_m)$. Thus, $m \geq \frac{n-1}{2}$. ■

Now, we can prove Theorem 2.4.1.

Proof: The algorithm described in Construction 2.3.2 yields a path of the length smaller than $2(|V|+1)$ for every simple, finite, undirected non-join graph (Proposition 2.3.3). Thus, the upper bound is $2(|V|+1)$. Proposition 2.4.7 shows that there is no algorithm that does better than $\frac{n-1}{2} = \frac{|V|-1}{2}$ for every graph Γ_n from the sequence of graphs $(\Gamma_n)_n$ described in Construction 2.4.3.

Thus, the worst-case length of a path generated by any algorithm lies between $\frac{|V|-1}{2}$ and $2(|V|+1)$. ■