Graphs

THIS BOOK provides an introduction to the research area of graph algorithms. — To start — in this chapter we review some graph–theoretic concepts.

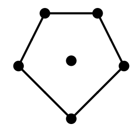

Figure 1.1: Kloks' Teacher, Professor Jaap Seidel, once told him that this was his favorite graph. (The point in the middle was added on only when he grew older.)

Definition 1.1. A graph G is an ordered pair of finite sets

$$G = (V, E).$$

The elements of the nonempty set V are called points — or vertices.[1] The set E is a subset of the unordered pairs of points

$$E \subseteq \{\{a, b\} \mid \{a, b\} \subseteq V\}$$

and the elements of E are called lines — or edges.[2]

[1] We have: one vertex, and two vertices.

[2] An edge is a set with two elements, which are two vertices.

When the sets V and E of a graph G are not clear from the context we denote them as $V(G)$ and $E(G)$. We require that $\underline{V \neq \varnothing}$.[3] A graph is called empty if $E = \varnothing$.

[3] A structure with $V = \varnothing$ is referred to as a 'null - graph.' (It is not a graph.) In English: 'null' = 'invalid.'

The two vertices of an edge are called the endpoints of that edge. An endpoint of an edge is said to be 'incident' with that edge. Two vertices x and y are adjacent if $\{x, y\} \in E$ and they are nonadjacent if $\{x, y\} \notin E$. When two vertices are adjacent — we say that they are — neighbors — of each other.

All graphs that we consider in this book are finite — that is — the set V is a nonempty finite set.

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022
T. Kloks, M. Xiao, *A Guide to Graph Algorithms*, https://doi.org/10.1007/978-981-16-6350-5_1

Definition 1.2. A set V is finite if

$$\exists_{k\in\mathbb{N}} |V| \leqslant k.$$

By the way, we use

$$\mathbb{N} = \{1, 2, \ldots\}.$$

The set $\mathbb{N}$ of the natural numbers is, by definition, countable but, it is not finite.

1.1 Isomorphic Graphs

Definition 1.3. Two graphs G_1 and G_2 are isomorphic if there exists a bijection $\pi : V(G_1) \to V(G_2)$ satisfying

$$\{x, y\} \in E(G_1) \quad \Leftrightarrow \quad \{\pi(x), \pi(y)\} \in E(G_2). \tag{1.1}$$

Exercise 1.1

Show that ‘being isomorphic’ is an equivalence relation — that is to say — show that

1. Every graph is isomorphic to itself
2. If G_1 is isomorphic to G_2 then G_2 is isomorphic to G_1
3. If G_1 is isomorphic to G_2 and G_2 is isomorphic to G_3 then G_1 is isomorphic to G_3.

1.2 Representing graphs

A graph can be represented by its adjacency matrix. Let G be a graph. The adjacency matrix A of G is a symmetric 0/1−matrix with rows and columns corresponding to the vertices of G. The elements $A_{x,y}$ of A are defined as follows.

$$\forall_{x\in V} \quad \forall_{y\in V} \quad A_{x,y} = \begin{cases} 1 & \text{if } \{x, y\} \in E(G), \text{ and} \\ 0 & \text{otherwise.} \end{cases} \tag{1.2}$$

— For example — the graph in Figure 1.1 on Page 1 has an adjacency matrix:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix} \tag{1.3}$$

1.3 Neighborhoods

Definition 1.4. The neighborhood $\mathsf{N}(x)$ of a vertex $x \in \mathsf{V}(\mathsf{G})$ is the set of vertices that are adjacent to x — that is

$$\mathsf{N}(x) = \{\, y \in \mathsf{V} \mid \{x, y\} \in \mathsf{E} \,\}. \tag{1.4}$$

When the graph G is not clear from the context we use $\mathsf{N}_{\mathsf{G}}(x)$ instead of $\mathsf{N}(x)$.

The degree of a vertex x is the number of its neighbors, ie [4]

$$d(x) = |\mathsf{N}(x)|. \tag{1.5}$$

We use the notation

$$\mathsf{N}[x] = \mathsf{N}(x) \cup \{x\} \tag{1.6}$$

to denote the 'closed neighborhood' of a vertex x. For a nonempty set W of vertices we write

$$\begin{aligned} \mathsf{N}(W) &= \left(\bigcup_{w \in W} \mathsf{N}(w) \right) \setminus W \\ &= \{\, y \mid y \in \mathsf{V} \setminus W \text{ and } \mathsf{N}(y) \cap W \neq \varnothing \,\}. \end{aligned} \tag{1.7}$$

We use $\mathsf{N}[W] = \mathsf{N}(W) \cup W$ to denote the closed neighborhood of W.

[4] A graph G is called regular if all vertices have the same degree.

Figure 1.2: The Petersen graph is regular with degree 3.

1.4 Connectedness

A graph is connected if one can walk from any point to any other point via the edges of the graph.[5]

[5] When I told my teacher AB, back in 1984, that I needed this property to finish my proof, he said: "That's called CONNECTED!"

Definition 1.5. A graph G is connected if $|V| = 1$ — or else — if for every partition — $\{A, B\}$ of V — there exists elements $a \in A$ and $b \in B$ satisfying $\{a, b\} \in E$.

A graph is disconnected if it is not connected.

A partition of a set is defined as follows.

Definition 1.6. A collection of sets $\{V_1, \ldots, V_t\}$ is a partition of a set V if

1. $t \geqslant 2$,
2. all $V_i \neq \varnothing$,
3. for all $i \neq j$, $V_i \cap V_j = \varnothing$, and
4. $\cup V_i = V$.

1.5 Induced Subgraphs

Let G be a graph. A subgraph of G is a graph H with[6]

$$V(H) \subseteq V(G) \quad \text{and} \quad E(H) \subseteq E(G). \tag{1.8}$$

[6] The graph H is a spanning subgraph of G if $V(H) = V(G)$ and $E(H) \subseteq E(G)$. When H is a spanning subgraph of G and also a tree it is called a spanning tree of G. Clearly, a graph can only have a spanning tree if it is connected.

Definition 1.7. Let G be a graph. Let $W \subseteq V(G)$ and let $W \neq \varnothing$. The subgraph of G induced by W is the graph H with

$$V(H) = W \quad \text{and}$$
$$E(H) = \{\{a, b\} \mid \{a, b\} \subseteq W \text{ and } \{a, b\} \in E(G)\}.$$

For a graph G and a nonempty set $W \subseteq V(G)$ we denote the subgraph of G induced by W as $G[W]$. We also write — where we use $V = V(G)$ —

$$G - W = G[V \setminus W], \quad \text{when } V \setminus W \neq \varnothing. \tag{1.9}$$

For a vertex x we write $G - x$ instead of $G - \{x\}$. For an edge $e = \{x, y\}$ we write $G - e$ for the graph with vertices $V(G)$ and edges $E(G) \setminus \{e\}$.

1.6 Paths and Cycles

Let G be a graph. A path[7] in G is a nonempty set of vertices that is ordered such that consecutive pairs are adjacent. A path has —by definition— at least one vertex. To denote a path P in G we use the notation

$$P = [x_1 \quad \cdots \quad x_t]. \tag{1.10}$$

— Here — the vertices of P are

$$V(P) = \{x_1, \cdots, x_t\} \subseteq V(G).$$

The edges of P are the pairs of vertices that are consecutive in the ordering. Edges in G — that connect vertices in P which are not consecutive — are called chords of P.

A path with n vertices has $n - 1$ edges.

The points x_1 and x_t are the endpoints of P and we also say that P runs between x_1 and x_t. We call a path P an $x{\sim}y$-path if x and y are the endpoints of P. The length of P is the number of edges in it — that is to say —

$$\ell(P) = |E(P)| = |V(P)| - 1.$$

Definition 1.8. For any two vertices x and y in a graph their distance is defined as

$$d(x, y) = \min\{\, |E(P)| \mid P \text{ is an } x{\sim}y\text{-path} \,\}. \tag{1.11}$$

[7] In a path all vertices are distinct.

In the case where $V(P) = V(G)$ and $E(P) = E(G)$ — the graph G is called a path.[8] We use a special notation to denote graphs that are paths with n vertices — namely —

$$\forall_{n \in \mathbb{N}} \quad P_n \text{ is the path with } n \text{ vertices}.$$

[8] A graph is a path if it contains a path to which it is isomorphic.

Figure 1.3: The figure shows a path with 4 vertices; that is, P_4.

Exercise 1.2

Design an algorithm to check whether a graph G given as input is a path.

A cycle[9] C in a graph G is a set of at least three vertices that is ordered

$$C = [c_1 \quad \cdots \quad c_t] \tag{1.12}$$

such that any two consecutive vertices are adjacent in G and — furthermore — also the first and last vertex of the sequence are adjacent.

[9] In a cycle all vertices are distinct.

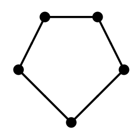

Figure 1.4: C_5: a cycle with 5 vertices

The edges of the cycle C are the consecutive pairs and the pair $\{c_1, c_t\}$. Edges

$$\{c_i, c_j\} \in E(G) \setminus E(C)$$

are called chords of C.

When $V(G) = V(C)$ and $E(G) = E(C)$ we say that G is a cycle. For graphs isomorphic to a cycle with n vertices we use the special notation

$$C_n \quad \text{is the cycle with } n \text{ vertices}.$$

Exercise 1.3

Show that a graph G is regular with all vertices of degree two, if and only if every component of G induces a cycle.

Exercise 1.4

Show that $\bar{C}_5$ and C_5 are isomorphic. Are there other cycles isomorphic to their complement?

1.7 Complements

Strawberries and cream complement each other nicely!

Let G be a graph. The complement of G is the graph $\bar{\mathsf{G}}$ with the following sets of vertices and edges.

$$V(\bar{G}) = V(G) \quad \text{and}$$
$$E(\bar{G}) = \{\{a, b\} \mid \{a, b\} \subseteq V(G) \quad \text{and} \quad \{a, b\} \notin E(G)\}.$$

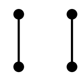

Figure 1.5: The figure shows $\bar{C}_4$, the complement of the 4-cycle. Notice that this graph is disconnected.

1.8 Components

UNIONS of connected graphs — without any additional edges between them — create new graphs. The constituent parts of the new graph are called its components.

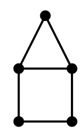

Figure 1.6: This is the house. It is the complement of P_5: $\bar{P}_5$.

Definition 1.9. A component of a graph G is a maximal set of vertices $W \subseteq V(G)$ such that $G[W]$ is connected.

Notice that, the set of components of a graph G forms a partition of $V(G)$ if and only if G is disconnected. Each component induces a connected subgraph of G.

1.8.1 Rem's Algorithm

Let G be a graph and let $V(G) = [n]$.[10] Rem's algorithm, Algorithm 1 computes a function $\delta : V(G) \to [n]$ satisfying

$$\forall_k \quad \delta(k) \leqslant k \quad \text{and}$$

(i) $\delta(k) = k$ if vertex k belongs to the component with number k

(ii) $\delta(k) < k$ if vertex k lies in the same component as vertex $\delta(k)$.

[10] $[n] = \{1, \ldots, n\}$

For the number of components we have

$$\#\text{ components of } \mathsf{G} = \#\{\, x \mid x \in V \quad \text{and} \quad \delta(x) = x \,\}. \quad (1.13)$$

Notice that when G is the empty graph [11] the identity function is a solution — it puts a vertex k in the component with number k, for $k \in [n]$.

[11] G is empty if $E(G) = \varnothing$.

Rem's algorithm initializes the function δ as the identity function and, as it — one by one — adds the edges to the graph, it updates δ. The objective is to decrease the number of applications needed for a vertex to find the representative vertex in its component.[12] The algorithm is optimal in the sense that δ decreases at every pass of the loop that starts in Line 11.

[12] That is, the number of times one has to apply δ before it gets constant.

Exercise 1.5

Prove the correctness of Rem's algorithm in Algorithm 1. Show that it runs in $O(n^2)$ time. Implement and test it on the graph shown in Figure 2.2 on Page 23 — that is the graph with — say 20 — disjoint triangles.

Exercise 1.6

Design an algorithm that — using Rem's algorithm as a subroutine — checks whether two vertices x and y are in the same component of a graph G.
Hint: Analyze what happens to their components if you add the edge $\{x, y\}$ to the graph.

Exercise 1.7

Suppose we measure the 'efficiency' of the algorithm by the maximal number of times one has to apply the function δ to a vertex to find its component's number. The efficiency of Rem's algorithm seems to depend on the order in which we add the edges. Is there a clever choice?
Hint: Suppose G is a path. Analyze different orderings in which to add the edges.

Algorithm 1: Rem's Algorithm

```
procedure Rem

    for i ∈ [n] do
        δ(i) ← i
    end for

    for {p, q} ∈ E(G) do
        p_0, q_0 ← p, q
        p_1, q_1 ← δ(p_0), δ(q_0)

        while p_1 ≠ q_1 do
            if p_1 < q_1 then
                δ(q_0) ← p_1
                q_0, q_1 ← q_1, δ(q_1)
            else
                δ(p_0) ← q_1
                p_0, p_1 ← p_1, δ(p_1)
            end if
        end while

    end for
end procedure
```

1.9 Separators

Definition 1.10. Let G be a graph.

1. A separator is a set $S \subset V$ such that $G - S$ is disconnected.
2. For two nonadjacent vertices a and b, a set S is a minimal $a\,|\,b$-separator if a and b are in different components of $G - S$ and no proper subset of S has that property.
3. A set $S \subset V(G)$ is a minimal separator if S is a minimal $a|b$-separator for some nonadjacent pair a and b.

— In a graph — the empty set is a minimal separator if and only if the graph is disconnected. If a minimal separator contains only one vertex that vertex is called a cutvertex.

Exercise 1.8

Let a and b be nonadjacent vertices in a graph. Prove that there is exactly one minimal $a|b$-separator contained in $N(a)$.

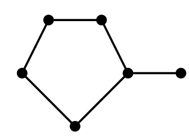

Figure 1.7: Notice that one minimal separator can be properly contained in another one. This is clear when G is disconnected. In this example, the cutvertex is properly contained in a minimal separator for the nonadjacent pair of its neighbors in the 5-cycle.

Exercise 1.9

Show that a set S is a minimal separator in a graph G if and only if $G - S$ has two components that have a neighbor of every vertex in S.

1.10 Trees

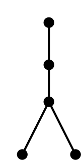

Figure 1.8: A tree

Definition 1.11. A connected graph T is a tree if it contains no cycles — that is — if no induced subgraph of T is a cycle.

Remark 1.12. Notice that Definition 1.11 implies that every connected induced subgraph of a tree is a tree.

Definition 1.13. A graph is a forest if each of its components induces a tree.

Exercise 1.10

Prove or disprove: A connected graph is a tree if and only if every minimal separator in it has cardinality 1. What are graphs in which every connected induced subgraph with at least three vertices has a cutvertex?

Theorem 1.14. *A connected graph is a tree if and only if every connected induced subgraph with at least two vertices has a vertex of degree one.* [13]

Proof. When a connected graph is not a tree then it contains a cycle, and so, the graph has an induced subgraph — the cycle — without vertex of degree 1.

Let T be a tree. When $|V(T)| = 1$ the only induced subgraph of T is T itself. Since T has no induced subgraph with at least two vertices, the condition is void (as it should be).

Now assume that $|V(T)| > 1$ and let T be connected. We show that T has two leaves. Let x be any vertex of T and let $C_1, \cdots, C_t$ be the components of $T - x$. If some component C_i has only one vertex, then this vertex has degree one, since its only neighbor is x.

If x has two neighbors in C_1, say y and z, then there is a path P in $T[C_1]$ connecting y and z. Then $V(P) \cup \{x\}$ induces a cycle in T.

By induction — on the number of vertices of T — we may assume that each C_i induces a tree with at least two vertices. By induction on the number of vertices — since $|C_i| > 1$ — $T[C_i]$ contains at least <u>two</u> leaves. At least one of those leaves is not adjacent to x — and so — there is a leaf in $T[C_i]$ which is a leaf in T.

To see that T has at least two leaves — observe that when $T - x$ has only one component that component contains a leaf

[13] In a tree T we call the vertices of degree at most one the <u>leaves</u> of T. Note that we have: 'one leaf' and 'two leaves.' (I did it wrong in my PhD thesis **;-(.**) In graphs, vertices of degree one are called <u>pendant</u> vertices. A pendant vertex is pendent (="dangling.") 'Pedant' means something entirely different.

and x itself is a leaf. When $T - x$ has at least two components — each component contains a leaf — and so there are at least two leaves.

This proves the theorem. $\square$

Notice that the proof of Theorem 1.14 shows that we can prune [14] leaves off a tree, until no vertex remains.

[14] 'To prune' means, to cut off branches from a plant or tree. In our case, we just pick leaves one by one.

Corollary 1.15. *A graph T is a tree if and only if it has an elimination order — that is an ordering of $V(T)$ — say*

$$[x_1 \quad \cdots \quad x_n]$$

such that

$$\forall_{1 \leqslant i \leqslant n} \quad G[V_i] \text{ is a tree and } x_i \text{ is a leaf in } G[V_i] \text{ where } V_i = \{x_i, \cdots, x_n\}. \tag{1.14}$$

Exercise 1.11

Design an algorithm that checks whether a graph given as input is a tree.

1.11 Bipartite Graphs

Definition 1.16. A graph G is bipartite if $|V(G)| = 1$ or else there is a partition of $V(G)$ — say $\{A, B\}$ — such that

$$\forall_{e \in E(G)} \quad e \cap A \neq \varnothing \quad \text{and} \quad e \cap B \neq \varnothing. \tag{1.15}$$

The two sets A and B in a partition as above are called color classes of G. Notice that these color classes are independent sets in G — that is — no two vertices in one color class are adjacent.

Exercise 1.12

Show that every tree is bipartite.

Exercise 1.13

A bipartite graph is complete bipartite if every pair of vertices in different color classes is adjacent. We denote a complete bipartite graph, with a and b vertices in its color classes, as $K_{a,b}$. What are the minimal separators in a complete bipartite graph?

Definition 1.17. Let G be a graph and let

$$S \subseteq V \quad \text{and} \quad S \neq \varnothing.$$

The set S is an independent set if no two vertices in S are adjacent.

We represent the maximal cardinality of an independent set in G by

$$\alpha(G).$$

Let C be some set — the elements of which are called colors. A coloring of a graph G is a map

$$V(G) \to C$$

with the property that the endpoints of any edge in G receive different colors of C.

Definition 1.18. The chromatic number of a graph G

$$\chi(G)$$

is the smallest number of colors needed to color the vertices such that adjacent vertices have different colors.

By definition a graph G is bipartite if and only if $\chi(G) \leqslant 2$.

Exercise 1.14

Design an algorithm that checks whether a graph is bipartite. Hint: It doesn't seem clever to use Theorem 1.19.

We have seen that trees are exactly those connected graphs that contain no cycles. Bipartite graphs are characterized via cycles as follows.

Theorem 1.19. *A graph is bipartite if and only if all cycles in it are even.*

Proof. Let G be bipartite and let $\{A, B\}$ be a partition of V such that all edges have one end in A and the other in B. Let $C = [c_1 \;\cdots\; c_t]$ be a cycle. Then the elements c_i alternate between A and B. This proves that

$$|V(C)| = |E(C)| = t \quad \text{is even}. \tag{1.16}$$

Let G be a graph and assume that all cycles in G are even. We may assume that G is connected. Start coloring G by assigning an arbitrary vertex r the color A. Define the set A as the set of vertices that are at even distance from r. Let

$$B = V \setminus A.$$

We claim that $\{A, B\}$ is a coloring of G. Assume that A contains two adjacent vertices x and y. Then a shortest path from r to x — together with a shortest path from r to y — plus the edge $\{x, y\}$ — contains an odd cycle.

This proves the theorem. □

1.12 Linegraphs

Definition 1.20. Let G be a graph with at least one edge. [15] The linegraph $L(G)$ of G is the graph

$$\begin{aligned} V(L) &= E(G) \quad \text{and} \\ E(L) &= \{\{e_1, e_2\} \mid \{e_1, e_2\} \subseteq E(G) \quad \text{and} \quad e_1 \cap e_2 \neq \varnothing\}. \end{aligned} \tag{1.17}$$

[15] When a graph is empty, its 'linegraph' would have no vertices, which makes it not a graph.

Exercise 1.15

Describe the linegraphs of trees (with at least one edge).

Notice that linegraphs are graphs without claws (see Figure 2.11). For any fixed graph H, a graph is H-free if it does not have an induced subgraph isomorphic to H.

1.13 Cliques and Independent Sets

Definition 1.21. A clique in a graph G is a nonempty set

$$C \subseteq V$$

such that any two vertices of C are adjacent.

— So — a clique in G is an independent set in $\bar{G}$ and vice versa.[16] Let $\omega(G)$ denote the maximal cardinality of a clique in G. Then

$$\omega(G) = \alpha(\bar{G}).$$

[16] An independent set is defined in Definition 1.17.

We call a clique with three vertices a triangle. Of course, bipartite graphs have no triangles, since those are odd cycles. It follows that $\omega(G) \leqslant 2$ whenever G is bipartite.

A graph is called a clique if every pair of its vertices are adjacent.[17] When G is a clique we have

$$\omega(G) = |V(G)| = \chi(G) \quad \text{and} \quad \alpha(G) = 1.$$

[17] For a clique with n vertices we use the notation K_n.

— Similarly — a graph is an independent set if it has no edges. When G is an independent set we have

$$\omega(G) = \chi(G) = 1 \quad \text{and} \quad \alpha(G) = |V(G)|.$$

1.14 On Notations

— To conclude this chapter — when there is no confusion possible we use — when a generic graph G underlies the discussion

$$n = |V(G)| \quad \text{and} \quad m = |E(G)|.$$

— Similarly — we use

$$V = V(G) \quad E = E(G) \quad \omega = \omega(G) \quad \alpha = \alpha(G) \quad \text{\&tc.}$$

whenever it is clear and convenient.[18]

[18] Perhaps I should have used v and e, instead of n and m, but this choice has the advantage that I am used to it!
I *should* have used G instead of V (and use the 'old' notation where G is synonymous with (G, E)), but, unfortunately, 'recent traditions' kept me back.

We freely abuse notation when it clarifies the text — for example — when S is some subset of vertices of a graph G then we use S also to denote $G[S]$ (see the Sidenote 2.4 on Page 30). The general rule applied removes uninformative variables and parentheses.[19]

[19] ...to improve readability; we don't want to compress the text. ;-)