Juho Hirvonen and Jukka Suomela:

Distributed Algorithms 2020

https://jukkasuomela.fi/da2020/ · March 14, 2025

Chapter 2
Graph-Theoretic Foundations

The study of distributed algorithms is closely related to graphs: we will interpret a computer network as a graph, and we will study computational problems related to this graph. In this section we will give a summary of the graph-theoretic concepts that we will use.

2.1 Terminology

A *simple undirected graph* is a pair $G = (V, E)$, where V is the set of *nodes* (*vertices*) and E is the set of *edges*. Each edge $e \in E$ is a 2-subset of nodes, that is, $e = \{u, v\}$ where $u \in V$, $v \in V$, and $u \neq v$. Unless otherwise mentioned, we assume that V is a non-empty finite set; it follows that E is a finite set. Usually, we will draw graphs using circles and lines—each circle represents a node, and a line that connects two nodes represents an edge.

2.1.1 Adjacency

If $e = \{u, v\} \in E$, we say that node u is *adjacent* to v, nodes u and v are *neighbors*, node u is *incident* to e, and edge e is also *incident* to u. If $e_1, e_2 \in E$, $e_1 \neq e_2$, and $e_1 \cap e_2 \neq \varnothing$ (i.e., e_1 and e_2 are distinct edges that share an endpoint), we say that e_1 is *adjacent* to e_2.

The *degree* of a node $v \in V$ in graph G is

$$\deg_G(v) = \Big|\Big\{\, u \in V : \{u, v\} \in E \,\Big\}\Big|.$$

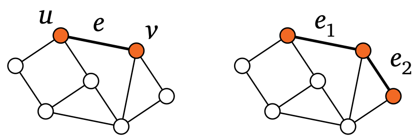

Figure 2.1: Node u is adjacent to node v. Nodes u and v are incident to edge e. Edge e_1 is adjacent to edge e_2.

That is, v has $\deg_G(v)$ neighbors; it is adjacent to $\deg_G(v)$ nodes and incident to $\deg_G(v)$ edges. A node $v \in V$ is *isolated* if $\deg_G(v) = 0$. Graph G is *k-regular* if $\deg_G(v) = k$ for each $v \in V$.

2.1.2 Subgraphs

Let $G = (V, E)$ and $H = (V_2, E_2)$ be two graphs. If $V_2 \subseteq V$ and $E_2 \subseteq E$, we say that H is a *subgraph* of G. If $V_2 = V$, we say that H is a *spanning subgraph* of G.

If $V_2 \subseteq V$ and $E_2 = \{\{u, v\} \in E : u \in V_2,\ v \in V_2\}$, we say that $H = (V_2, E_2)$ is an *induced subgraph*; more specifically, H is the subgraph of G induced by the set of nodes V_2.

If $E_2 \subseteq E$ and $V_2 = \bigcup E_2$, we say that H is an *edge-induced subgraph*; more specifically, H is the subgraph of G induced by the set of edges E_2.

2.1.3 Walks

A *walk* of length ℓ from node v_0 to node v_ℓ is an alternating sequence

$$w = (v_0, e_1, v_1, e_2, v_2, \dots, e_\ell, v_\ell)$$

where $v_i \in V$, $e_i \in E$, and $e_i = \{v_{i-1}, v_i\}$ for all i; see Figure 2.2. The walk is *empty* if $\ell = 0$. We say that walk w *visits* the nodes $v_0, v_1, \dots, v_\ell$, and it *traverses* the edges $e_1, e_2, \dots, e_\ell$. In general, a walk may visit the same node more than once and it may traverse the same edge more than once. A *non-backtracking walk* does not traverse the same edge twice consecutively, that is, $e_{i-1} \neq e_i$ for all i. A *path* is a walk that visits each

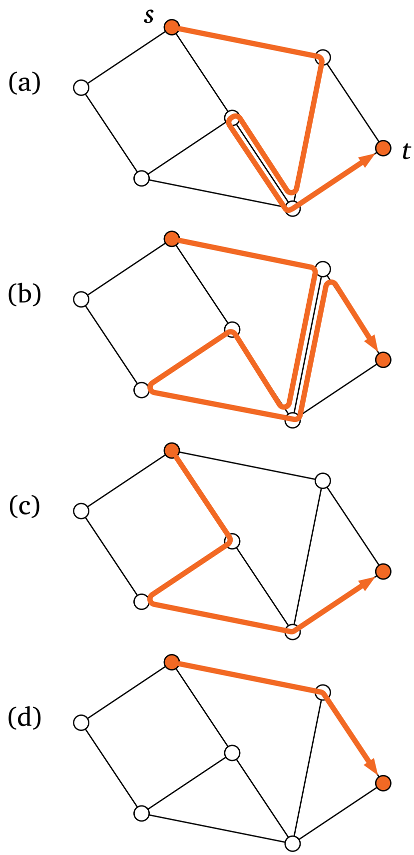

Figure 2.2: (a) A walk of length 5 from s to t. (b) A non-backtracking walk. (c) A path of length 4. (d) A path of length 2; this is a shortest path and hence $\mathrm{dist}_G(s,t) = 2$.

(a)

(b)

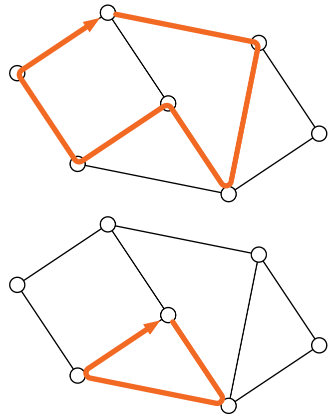

Figure 2.3: (a) A cycle of length 6. (b) A cycle of length 3; this is a shortest cycle and hence the girth of the graph is 3.

node at most once, that is, $v_i \neq v_j$ for all $0 \leq i < j \leq \ell$. A walk is *closed* if $v_0 = v_\ell$. A *cycle* is a non-empty closed walk with $v_i \neq v_j$ and $e_i \neq e_j$ for all $1 \leq i < j \leq \ell$; see Figure 2.3. Note that the length of a cycle is at least 3.

2.1.4 Connectivity and Distances

For each graph $G = (V, E)$, we can define a relation $\rightsquigarrow$ on V as follows: $u \rightsquigarrow v$ if there is a walk from u to v. Clearly $\rightsquigarrow$ is an equivalence relation. Let $C \subseteq V$ be an equivalence class; the subgraph induced by C is called a *connected component* of G.

If u and v are in the same connected component, there is at least one *shortest path* from u to v, that is, a path from u to v of the smallest possible length. Let ℓ be the length of a shortest path from u to v; we define that the *distance* between u and v in G is $\mathrm{dist}_G(u, v) = \ell$. If u and v are not in the same connected component, we define $\mathrm{dist}_G(u, v) = \infty$. Note that $\mathrm{dist}_G(u, u) = 0$ for any node u.

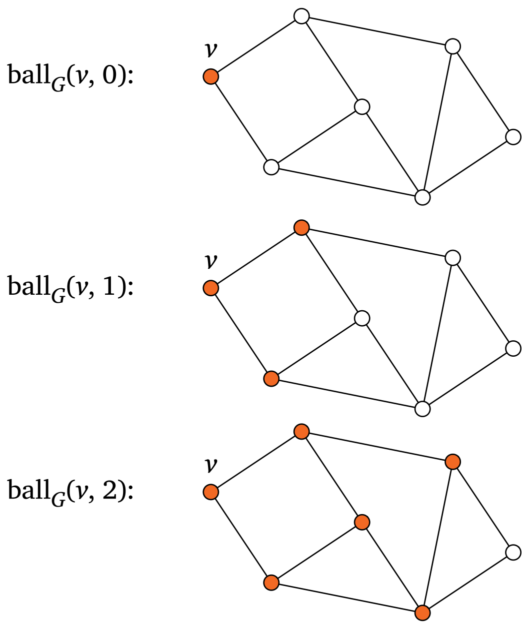

Figure 2.4: Neighborhoods.

For each node v and for a non-negative integer r, we define the *radius-r neighborhood* of v as follows (see Figure 2.4):

$$\mathrm{ball}_G(v,r) = \{ u \in V : \mathrm{dist}_G(u,v) \le r \}.$$

A graph is *connected* if it consists of one connected component. The *diameter* of graph G, in notation $\mathrm{diam}(G)$, is the length of a longest shortest path, that is, the maximum of $\mathrm{dist}_G(u,v)$ over all $u,v \in V$; we have $\mathrm{diam}(G) = \infty$ if the graph is not connected.

The *girth* of graph G is the length of a shortest cycle in G. If the graph does not have any cycles, we define that the girth is ∞; in that case we say that G is *acyclic*.

A *tree* is a connected, acyclic graph. If $T = (V,E)$ is a tree and $u,v \in V$, then there exists precisely one path from u to v. An acyclic graph is also known as a *forest*—in a forest each connected component is a tree. A *pseudotree* has at most one cycle, and in a *pseudoforest* each connected component is a pseudotree.

A *path graph* is a graph that consists of one path, and a *cycle graph* is a graph that consists of one cycle. Put otherwise, a path graph is a tree in which all nodes have degree at most 2, and a cycle graph is a 2-regular pseudotree. Note that any graph of maximum degree 2 consists of disjoint paths and cycles, and any 2-regular graph consists of disjoint cycles.

2.1.5 Isomorphism

An *isomorphism* from graph $G_1 = (V_1,E_1)$ to graph $G_2 = (V_2,E_2)$ is a bijection $f : V_1 \to V_2$ that preserves adjacency: $\{u,v\} \in E_1$ if and only if $\{f(u), f(v)\} \in E_2$. If an isomorphism from G_1 to G_2 exists, we say that G_1 and G_2 are isomorphic.

If G_1 and G_2 are isomorphic, they have the same structure; informally, G_2 can be constructed by renaming the nodes of G_1 and vice versa.

2.2 Packing and Covering

A subset of nodes $X \subseteq V$ is

(a) an *independent set* if each edge has at most one endpoint in X, that is, $|e \cap X| \leq 1$ for all $e \in E$,

(b) a *vertex cover* if each edge has at least one endpoint in X, that is, $e \cap X \neq \varnothing$ for all $e \in E$,

(c) a *dominating set* if each node $v \notin X$ has at least one neighbor in X, that is, $\mathrm{ball}_G(v,1) \cap X \neq \varnothing$ for all $v \in V$.

A subset of edges $X \subseteq E$ is

(d) a *matching* if each node has at most one incident edge in X, that is, $\{t,u\} \in X$ and $\{t,v\} \in X$ implies $u = v$,

(e) an *edge cover* if each node has at least one incident edge in X, that is, $\bigcup X = V$,

(f) an *edge dominating set* if each edge $e \notin X$ has at least one neighbor in X, that is, $e \cap \left(\bigcup X\right) \neq \varnothing$ for all $e \in E$.

See Figure 2.5 for illustrations.

Independent sets and matchings are examples of *packing problems*—intuitively, we have to "pack" elements into set X while avoiding conflicts. Packing problems are *maximization problems*. Typically, it is trivial to find a feasible solution (for example, an empty set), but it is more challenging to find a large solution.

Vertex covers, edge covers, dominating sets, and edge dominating sets are examples of *covering problems*—intuitively, we have to find a set X that "covers" the relevant parts of the graph. Covering problems are *minimization problems*. Typically, it is trivial to find a feasible solution if it exists (for example, the set of all nodes or all edges), but it is more challenging to find a small solution.

The following terms are commonly used in the context of maximization problems; it is important not to confuse them:

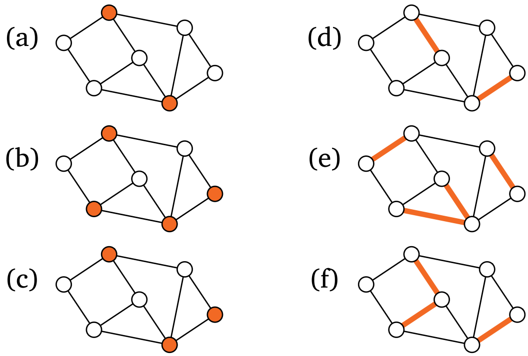

Figure 2.5: Packing and covering problems; see Section 2.2.

(a) ***maximal***: a maximal solution is not a proper subset of another feasible solution,

(b) ***maximum***: a maximum solution is a solution of the largest possible cardinality.

Similarly, in the context of minimization problems, analogous terms are used:

(a) ***minimal***: a minimal solution is not a proper superset of another feasible solution,

(b) ***minimum***: a minimum solution is a solution of the smallest possible cardinality.

Using this convention, we can define the terms *maximal independent set*, *maximum independent set*, *maximal matching*, *maximum matching*, *minimal vertex cover*, *minimum vertex cover*, etc.

For example, Figure 2.5a shows a maximal independent set: it is not possible to greedily extend the set by adding another element. However, it is not a maximum independent set: there exists an independent set of size 3. Figure 2.5d shows a matching, but it is not a maximal matching, and therefore it is not a maximum matching either.

Typically, maximal and minimal solutions are easy to find—you can apply a greedy algorithm. However, maximum and minimum solutions can be very difficult to find—many of these problems are NP-hard optimization problems.

A *minimum maximal matching* is precisely what the name suggests: it is a maximal matching of the smallest possible cardinality. We can define a *minimum maximal independent set*, etc., in an analogous manner.

2.3 Labelings and Partitions

We will often encounter functions of the form

$$f : V \to \{1, 2, \dots, k\}.$$

There are two interpretations that are often helpful:

(i) Function f assigns a *label* $f(v)$ to each node $v \in V$. Depending on the context, the labels can be interpreted as colors, time slots, etc.

(ii) Function f is a *partition* of V. More specifically, f defines a partition $V = V_1 \cup V_2 \cup \dots \cup V_k$ where $V_i = f^{-1}(i) = \{ v \in V : f(v) = i \}$.

Similarly, we can study a function of the form

$$f : E \to \{1, 2, \dots, k\}$$

and interpret it either as a labeling of edges or as a partition of E.

Many graph problems are related to such functions. We say that a function $f : V \to \{1, 2, \dots, k\}$ is

(a) a *proper vertex coloring* if $f^{-1}(i)$ is an independent set for each i,

(b) a *weak coloring* if each non-isolated node u has a neighbor v with $f(u) \neq f(v)$,

(c) a *domatic partition* if $f^{-1}(i)$ is a dominating set for each i.

A function $f : E \to \{1, 2, \dots, k\}$ is

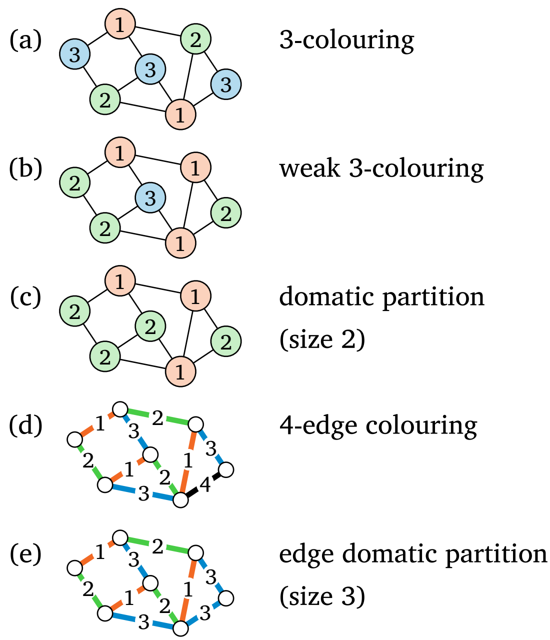

Figure 2.6: Partition problems; see Section 2.3.

(d) a *proper edge coloring* if $f^{-1}(i)$ is a matching for each i,

(e) an *edge domatic partition* if $f^{-1}(i)$ is an edge dominating set for each i.

See Figure 2.6 for illustrations.

Usually, the term *coloring* refers to a proper vertex coloring, and the term *edge coloring* refers to a proper edge coloring. The value of k is the *size* of the coloring or the *number of colors*. We will use the term *k-coloring* to refer to a proper vertex coloring with k colors; the term *k-edge coloring* is defined in an analogous manner.

A graph that admits a 2-coloring is a *bipartite graph*. Equivalently, a bipartite graph is a graph that does not have an odd cycle.

Graph coloring is typically interpreted as a minimization problem. It is easy to find a proper vertex coloring or a proper edge coloring if we can use arbitrarily many colors; however, it is difficult to find an *optimal* coloring that uses the smallest possible number of colors.

On the other hand, domatic partitions are a maximization problem. It is trivial to find a domatic partition of size 1; however, it is difficult to find an *optimal* domatic partition with the largest possible number of disjoint dominating sets.

2.4 Factors and Factorizations

Let $G = (V, E)$ be a graph, let $X \subseteq E$ be a set of edges, and let $H = (U, X)$ be the subgraph of G induced by X. We say that X is a *d-factor* of G if $U = V$ and $\deg_H(v) = d$ for each $v \in V$.

Equivalently, X is a d-factor if X induces a spanning d-regular subgraph of G. Put otherwise, X is a d-factor if each node $v \in V$ is incident to exactly d edges of X.

A function $f : E \to \{1, 2, \ldots, k\}$ is a *d-factorization* of G if $f^{-1}(i)$ is a d-factor for each i. See Figure 2.7 for examples.

We make the following observations:

(a) A 1-factor is a maximum matching. If a 1-factor exists, a maximum matching is a 1-factor.

(b) A 1-factorization is an edge coloring.

(c) The subgraph induced by a 2-factor consists of disjoint cycles.

A 1-factor is also known as a *perfect matching*.

2.5 Approximations

So far we have encountered a number of maximization problems and minimization problems. More formally, the definition of a maximization problem consists of two parts: a set of *feasible solutions* $\mathscr{S}$ and an *objective*

(a)

(b)

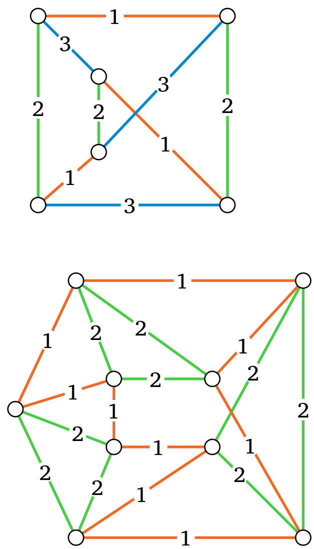

Figure 2.7: (a) A 1-factorization of a 3-regular graph. (b) A 2-factorization of a 4-regular graph.

function $g\colon \mathscr{S} \to \mathbb{R}$. In a maximization problem, the goal is to find a feasible solution $X \in \mathscr{S}$ that maximizes $g(X)$. A minimization problem is analogous: the goal is to find a feasible solution $X \in \mathscr{S}$ that minimizes $g(X)$.

For example, the problem of finding a maximum matching for a graph G is of this form. The set of feasible solutions $\mathscr{S}$ consists of all matchings in G, and we simply define $g(M) = |M|$ for each matching $M \in \mathscr{S}$.

As another example, the problem of finding an optimal coloring is a minimization problem. The set of feasible solutions $\mathscr{S}$ consists of all proper vertex colorings, and $g(f)$ is the number of colors in $f \in \mathscr{S}$.

Often, it is infeasible or impossible to find an optimal solution; hence we resort to approximations. Given a maximization problem $(\mathscr{S}, g)$, we say that a solution X is an *α-approximation* if $X \in \mathscr{S}$, and we have $\alpha g(X) \geq g(Y)$ for all $Y \in \mathscr{S}$. That is, X is a feasible solution, and the size of X is within factor α of the optimum.

Similarly, if $(\mathscr{S}, g)$ is a minimization problem, we say that a solution X is an α-approximation if $X \in \mathscr{S}$, and we have $g(X) \leq \alpha g(Y)$ for all $Y \in \mathscr{S}$. That is, X is a feasible solution, and the size of X is within factor α of the optimum.

Note that we follow the convention that the approximation ratio α is always at least 1, both in the case of minimization problems and maximization problems. Other conventions are also used in the literature.

2.6 Directed Graphs and Orientations

Unless otherwise mentioned, all graphs that we encounter are undirected. However, we will occasionally need to refer to so-called orientations, and hence we need to introduce some terminology related to directed graphs.

A *directed graph* is a pair $G = (V, E)$, where V is the set of nodes and E is the set of *directed edges*. Each edge $e \in E$ is a pair of nodes, that is, $e = (u, v)$ where $u, v \in V$. Put otherwise, $E \subseteq V \times V$.

Intuitively, an edge (u,v) is an "arrow" that points from node u to node v; it is an *outgoing edge* for u and an *incoming edge* for v. The *outdegree* of a node $v \in V$, in notation $\operatorname{outdegree}_G(v)$, is the number of outgoing edges, and the *indegree* of the node, $\operatorname{indegree}_G(v)$, is the number of incoming edges.

Now let $G = (V,E)$ be a graph and let $H = (V,E')$ be a directed graph with the same set of nodes. We say that H is an *orientation* of G if the following holds:

(a) For each $\{u,v\} \in E$ we have either $(u,v) \in E'$ or $(v,u) \in E'$, but not both.

(b) For each $(u,v) \in E'$ we have $\{u,v\} \in E$.

Put otherwise, in an orientation of G we have simply chosen an arbitrary direction for each undirected edge of G. It follows that

$$\operatorname{indegree}_H(v) + \operatorname{outdegree}_H(v) = \deg_G(v)$$

for all $v \in V$.

2.7 Quiz

Construct a simple undirected graph $G = (V,E)$ with the following property: If you take any set X that is a maximal independent set of G, then X is not a minimum dominating set of G.

Present the graph in the set formalism by listing the sets of nodes and edges. For example, a cycle on three nodes can be encoded as $V = \{1,2,3\}$ and $E = \{\{1,2\},\{2,3\},\{3,1\}\}$.

2.8 Exercises

Exercise 2.1 (independence and vertex covers)**.** Let $I \subseteq V$ and define $C = V \setminus I$. Show that

(a) if I is an independent set then C is a vertex cover and vice versa,

(b) if I is a maximal independent set then C is a minimal vertex cover and vice versa,

(c) if I is a maximum independent set then C is a minimum vertex cover and vice versa,

(d) it is possible that C is a 2-approximation of minimum vertex cover but I is not a 2-approximation of maximum independent set,

(e) it is possible that I is a 2-approximation of maximum independent set but C is not a 2-approximation of minimum vertex cover.

Exercise 2.2 (matchings)**.** Show that

(a) any maximal matching is a 2-approximation of a maximum matching,

(b) any maximal matching is a 2-approximation of a minimum maximal matching,

(c) a maximal independent set is not necessarily a 2-approximation of maximum independent set,

(d) a maximal independent set is not necessarily a 2-approximation of minimum maximal independent set.

Exercise 2.3 (matchings and vertex covers)**.** Let M be a maximal matching, and let $C = \bigcup M$, i.e., C consists of all endpoints of matched edges. Show that

(a) C is a 2-approximation of a minimum vertex cover,

(b) C is not necessarily a 1.999-approximation of a minimum vertex cover.

Would you be able to improve the approximation ratio if M was a minimum maximal matching?

Exercise 2.4 (independence and domination)**.** Show that

(a) a maximal independent set is a minimal dominating set,

(b) a minimal dominating set is not necessarily a maximal independent set,

(c) a minimum maximal independent set is not necessarily a minimum dominating set.

Exercise 2.5 (graph colorings and partitions)**.** Show that

(a) a weak 2-coloring always exists,

(b) a domatic partition of size 2 does not necessarily exist,

(c) if a domatic partition of size 2 exists, then a weak 2-coloring is a domatic partition of size 2,

(d) a weak 2-coloring is not necessarily a domatic partition of size 2.

Show that there are 2-regular graphs with the following properties:

(e) any 3-coloring is a domatic partition of size 3,

(f) no 3-coloring is a domatic partition of size 3.

Assume that G is a graph of maximum degree Δ; show that

(g) there exists a $(\Delta + 1)$-coloring,

(h) a Δ-coloring does not necessarily exist.

Exercise 2.6 (isomorphism)**.** Construct non-empty 3-regular connected graphs G and H such that G and H have the same number of nodes and G and H are *not* isomorphic. Just giving a construction is not sufficient —you have to *prove* that G and H are not isomorphic.

⋆ **Exercise 2.7** (matchings and edge domination)**.** Show that

(a) a maximal matching is a minimal edge dominating set,

(b) a minimal edge dominating set is not necessarily a maximal matching,

(c) a minimum maximal matching is a minimum edge dominating set,

(d) any maximal matching is a 2-approximation of a minimum edge dominating set.

▹ *hint A*

⋆ **Exercise 2.8** (Petersen 1891)**.** Show that any $2d$-regular graph $G = (V, E)$ has an orientation $H = (V, E')$ such that

$$\text{indegree}_H(v) = \text{outdegree}_H(v) = d$$

for all $v \in V$. Show that any $2d$-regular graph has a 2-factorization.

2.9 Bibliographic Notes

The connection between maximal matchings and approximations of vertex covers (Exercise 2.3) is commonly attributed to Gavril and Yannakakis—see, e.g., Papadimitriou and Steiglitz [4]. The connection between minimum maximal matchings and minimum edge dominating sets (Exercise 2.7) is due to Allan and Laskar [1] and Yannakakis and Gavril [7]. Exercise 2.8 is a 120-year-old result due to Petersen [5]. The definition of a weak coloring is from Naor and Stockmeyer [3].

Diestel's book [2] is a good source for graph-theoretic background, and Vazirani's book [6] provides further information on approximation algorithms.

2.10 Bibliography

[1] Robert B. Allan and Renu Laskar. On domination and independent domination numbers of a graph. *Discrete Mathematics*, 23(2):73–76, 1978. doi:10.1016/0012-365X(78)90105-X.

[2] Reinhard Diestel. *Graph Theory*. Springer, 3rd edition, 2005. URL: http://diestel-graph-theory.com/.

[3] Moni Naor and Larry Stockmeyer. What can be computed locally? *SIAM Journal on Computing*, 24(6):1259–1277, 1995. doi:10.1137/S0097539793254571.

[4] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications, Inc., 1998.

[5] Julius Petersen. Die Theorie der regulären graphs. *Acta Mathematica*, 15(1):193–220, 1891. doi:10.1007/BF02392606.

[6] Vijay V. Vazirani. *Approximation Algorithms*. Springer, 2001.

[7] Mihalis Yannakakis and Fanica Gavril. Edge dominating sets in graphs. *SIAM Journal on Applied Mathematics*, 38(3):364–372, 1980. doi:10.1137/0138030.

2.11 Hints

A. Assume that D is an edge dominating set; show that you can construct a maximal matching M with $|M| \leq |D|$.