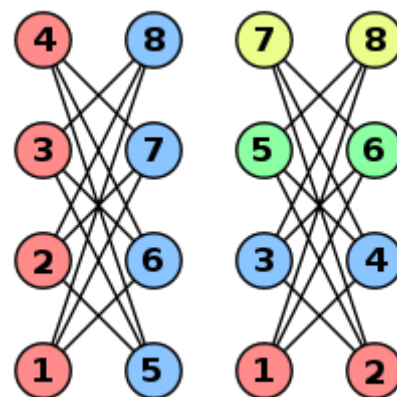


Greedy coloring

In the study of [graph coloring](#) problems in [mathematics](#) and [computer science](#), a **greedy coloring** or **sequential coloring**^[1] is a coloring of the [vertices](#) of a [graph](#) formed by a [greedy algorithm](#) that considers the [vertices](#) of the graph in sequence and assigns each vertex its [first available](#) color. Greedy colorings can be found in linear time, but they do not in general use the minimum number of colors possible.

Different choices of the sequence of vertices will typically produce different colorings of the given graph, so much of the study of greedy colorings has concerned how to find a good ordering. There always exists an ordering that produces an optimal coloring, but although such orderings can be found for many special classes of graphs, they are hard to find in general. Commonly used strategies for vertex ordering involve placing higher-degree vertices earlier than lower-degree vertices, or choosing vertices with fewer available colors in preference to vertices that are less constrained.

Variations of greedy coloring choose the colors in an [online manner](#), without any knowledge of the structure of the uncolored part of the graph, or choose other colors than the first available in order to reduce the total number of colors. Greedy coloring algorithms have been applied to [scheduling](#) and [register allocation](#) problems, the analysis of combinatorial games, and the proofs of other mathematical results including [Brooks' theorem](#) on the relation between coloring and degree. Other concepts in graph theory derived from greedy colorings include the [Grundy number](#) of a graph (the largest number of colors that can be found by a greedy coloring), and the [well-colored graphs](#), graphs for which all greedy colorings use the same number of colors.



Two greedy colorings of the same [crown graph](#) using different vertex orders. The right example generalises to 2-colorable graphs with n vertices, where the greedy algorithm expends $n/2$ colors.

Contents

[Algorithm](#)

[Choice of ordering](#)

[Good orders](#)

[Bad orders](#)

[Graphs for which order is irrelevant](#)

[Degeneracy](#)

[Adaptive orders](#)

[Alternative color selection schemes](#)

[Online selection](#)

[Parsimonous coloring](#)

[Applications](#)

[Notes](#)

[References](#)

Algorithm

The greedy coloring for a given vertex ordering can be computed by an algorithm that runs in linear time. The algorithm processes the vertices in the given ordering, assigning a color to each one as it is processed. The colors may be represented by the numbers $0, 1, 2, \dots$ and each vertex is given the color with the smallest number that is not already used by one of its neighbors. To find the smallest available color, one may use an array to count the number of neighbors of each color (or alternatively, to represent the set of colors of neighbors), and then scan the array to find the index of its first zero.^[2]

In Python, the algorithm can be expressed as:

```
def first_available(color_list):
    """Return smallest non-negative integer not in the given list of colors."""
    color_set = set(color_list)
    count = 0
    while True:
        if count not in color_set:
            return count
        count += 1

def greedy_color(G, order):
    """Find the greedy coloring of G in the given order.
    The representation of G is assumed to be like https://www.python.org/doc/essays/graphs/
    in allowing neighbors of a node/vertex to be iterated over by "for w in G[node]".
    The return value is a dictionary mapping vertices to their colors."""
    color = dict()
    for node in order:
        used_neighbour_colors = [color[nbr] for nbr in G[node] if nbr in color]
        color[node] = first_available(used_neighbour_colors)
    return color
```

The `first_available` subroutine takes time proportional to the length of its argument list, because it performs two loops, one over the list itself and one over a list of counts that has the same length. The time for the overall coloring algorithm is dominated by the calls to this subroutine. Each edge in the graph contributes to only one of these calls, the one for the endpoint of the edge that is later in the vertex ordering. Therefore, the sum of the lengths of the argument lists to `first_available`, and the total time for the algorithm, are proportional to the number of edges in the graph.^[2]

An alternative algorithm, producing the same coloring,^[3] is to choose the sets of vertices with each color, one color at a time. In this method, each color class C is chosen by scanning through the vertices in the given ordering. When this scan encounters an uncolored vertex v that has no neighbor in C , it adds v to C . In this way, C becomes a maximal independent set among the vertices that were not already assigned smaller colors. The algorithm repeatedly finds color classes in this way until all vertices are colored. However, it involves making multiple scans of the graph, one scan for each color class, instead of the method outlined above which uses only a single scan.^[4]

Choice of ordering

Different orderings of the vertices of a graph may cause the greedy coloring to use different numbers of colors, ranging from the optimal number of colors to, in some cases, a number of colors that is proportional to the number of vertices in the graph. For instance, a crown graph (a graph formed from two disjoint sets of $n/2$ vertices $\{a_1, a_2, \dots\}$ and $\{b_1, b_2, \dots\}$ by connecting a_i to b_j whenever $i \neq j$) can be a particularly bad case for greedy coloring. With the vertex ordering $a_1, b_1, a_2, b_2, \dots$, a greedy coloring will use $n/2$ colors, one color for each pair (a_i, b_i) . However, the optimal number of colors for this graph is two, one color for the

vertices a_i and another for the vertices b_i .^[5] There also exist graphs such that with high probability a randomly chosen vertex ordering leads to a number of colors much larger than the minimum.^[6] Therefore, it is of some importance in greedy coloring to choose the vertex ordering carefully.

Good orders

The vertices of any graph may always be ordered in such a way that the greedy algorithm produces an optimal coloring. For, given any optimal coloring, one may order the vertices by their colors. Then when one uses a greedy algorithm with this order, the resulting coloring is automatically optimal.^[7] However, because optimal graph coloring is NP-complete, any subproblem that would allow this problem to be solved quickly, including finding an optimal ordering for greedy coloring, is NP-hard.^[8]

In interval graphs and chordal graphs, if the vertices are ordered in the reverse of a perfect elimination ordering, then the earlier neighbors of every vertex will form a clique. This property causes the greedy coloring to produce an optimal coloring, because it never uses more colors than are required for each of these cliques. An elimination ordering can be found in linear time, when it exists.^[9]

More strongly, any perfect elimination ordering is *hereditarily optimal*, meaning that it is optimal both for the graph itself and for all of its induced subgraphs. The perfectly orderable graphs (which include chordal graphs, comparability graphs, and distance-hereditary graphs) are defined as the graphs that have a hereditarily optimal ordering.^[10] Recognizing perfectly orderable graphs is also NP-complete.^[11]

Bad orders

The number of colors produced by the greedy coloring for the worst ordering of a given graph is called its Grundy number.^[12] Just as finding a good vertex ordering for greedy coloring is difficult, so is finding a bad vertex ordering. It is NP-complete to determine, for a given graph G and number k , whether there exists an ordering of the vertices of G that causes the greedy algorithm to use k or more colors. In particular, this means that it is difficult to find the worst ordering for G .^[12]

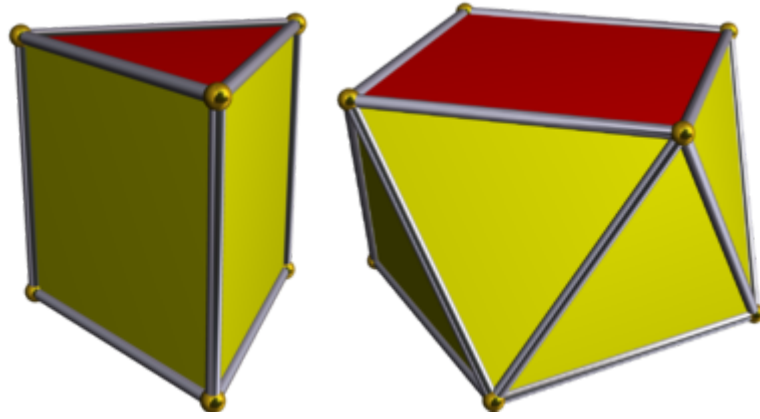
Graphs for which order is irrelevant

The well-colored graphs are the graphs for which all vertex colorings produce the same number of colors. This number of colors, in these graphs, equals both the chromatic number and the Grundy number.^[12] They include the cographs, which are exactly the graphs in which all induced subgraphs are well-colored.^[13] However, it is co-NP-complete to determine whether a graph is well-colored.^[12]

If a random graph is drawn from the Erdős–Rényi model with constant probability of including each edge, then any vertex ordering that is chosen independently of the graph edges leads to a coloring whose number of colors is close to twice the optimal value, with high probability. It remains unknown whether there is any polynomial time method for finding significantly better colorings of these graphs.^[3]

Degeneracy

Because optimal vertex orderings are hard to find, heuristics have been used that attempt to reduce the number of colors while not guaranteeing an optimal number of colors. A commonly used ordering for greedy coloring is to choose a vertex v of minimum degree, order the subgraph with v removed recursively, and then place v last in the ordering. The largest degree of a removed vertex that this algorithm encounters is called the degeneracy of the graph, denoted d . In the context of greedy coloring, the same ordering strategy is also called



The triangular prism and square antiprism, graphs whose greedy colorings using the degeneracy ordering give larger-than-optimal numbers of colors

the smallest last ordering.^[14] This vertex ordering, and the degeneracy, may be computed in linear time.^[15] It can be viewed as an improved version of an earlier vertex ordering method, the largest-first ordering, which sorts the vertices in descending order by their degrees.^[16]

With the degeneracy ordering, the greedy coloring will use at most $d + 1$ colors. This is because, when colored, each vertex will have at most d already-colored neighbors, so one of the first $d + 1$ colors will be free for it to use.^[17] Greedy coloring with the degeneracy ordering can find optimal colorings for certain classes of graphs, including trees, pseudoforests, and crown graphs.^[18] Markossian, Gasparian & Reed (1996) define a graph G to be β -perfect if, for G and every induced subgraph of G , the chromatic number equals the degeneracy plus one. For these graphs, the greedy algorithm with the degeneracy ordering is always optimal.^[19] Every β -perfect graph must be an even-hole-free graph, because even cycles have chromatic number two and degeneracy two, not matching the equality in the definition of β -perfect graphs. If a graph and its complement graph are both even-hole-free, they are both β -perfect. The graphs that are both perfect graphs and β -perfect graphs are exactly the chordal graphs. On even-hole-free graphs more generally, the degeneracy ordering approximates the optimal coloring to within at most twice the optimal number of colors; that is, its approximation ratio is 2.^[20] On unit disk graphs its approximation ratio is 3.^[21] The triangular prism is the smallest graph for which one of its degeneracy orderings leads to a non-optimal coloring, and the square antiprism is the smallest graph that cannot be optimally colored using any of its degeneracy orderings.^[18]

Adaptive orders

Brélaz (1979) proposes a strategy, called DSatur, for vertex ordering in greedy coloring that interleaves the construction of the ordering with the coloring process. In his version of the greedy coloring algorithm, the next vertex to color at each step is chosen as the one with the largest number of distinct colors in its neighborhood. In case of ties, a vertex of maximal degree in the subgraph of uncolored vertices is chosen from the tied vertices. By keeping track of the sets of neighboring colors and their cardinalities at each step, it is possible to implement this method in linear time.^[22]

This method can find the optimal colorings for bipartite graphs,^[23] all cactus graphs, all wheel graphs, all graphs on at most six vertices, and almost every k -colorable graph.^[24] Although Lévêque & Maffray (2005) originally claimed that this method finds optimal colorings for the Meyniel graphs, they later found a counterexample to this claim.^[25]

Alternative color selection schemes

It is possible to define variations of the greedy coloring algorithm in which the vertices of the given graph are colored in a given sequence but in which the color chosen for each vertex is not necessarily the first available color. These include methods in which the uncolored part of the graph is unknown to the algorithm, or in which the algorithm is given some freedom to make better coloring choices than the basic greedy algorithm would.

Online selection

Alternative color selection strategies have been studied within the framework of online algorithms. In the online graph-coloring problem, vertices of a graph are presented one at a time in an arbitrary order to a coloring algorithm; the algorithm must choose a color for each vertex, based only on the colors of and adjacencies among already-processed vertices. In this context, one measures the quality of a color selection strategy by its competitive ratio, the ratio between the number of colors it uses and the optimal number of colors for the given graph.^[26]

If no additional restrictions on the graph are given, the optimal competitive ratio is only slightly sublinear.^[27] However, for interval graphs, a constant competitive ratio is possible,^[28] while for bipartite graphs and sparse graphs a logarithmic ratio can be achieved. Indeed, for sparse graphs, the standard greedy coloring strategy of choosing the first available color achieves this competitive ratio, and it is possible to prove a matching lower bound on the competitive ratio of any online coloring algorithm.^[26]

Parsimonous coloring

A *parsimonious coloring*, for a given graph and vertex ordering, has been defined to be a coloring produced by a greedy algorithm that colors the vertices in the given order, and only introduces a new color when all previous colors are adjacent to the given vertex, but can choose which color to use (instead of always choosing the smallest) when it is able to re-use an existing color. The *ordered chromatic number* is the smallest number of colors that can be obtained for the given ordering in this way, and the *ochromatic number* is the largest ordered chromatic number among all vertex colorings of a given graph. Despite its different definition, the ochromatic number always equals the Grundy number.^[29]

Applications

Because it is fast and in many cases can use few colors, greedy coloring can be used in applications where a good but not optimal graph coloring is needed. One of the early applications of the greedy algorithm was to problems such as course scheduling, in which a collection of tasks must be assigned to a given set of time slots, avoiding incompatible tasks being assigned to the same time slot.^[4] It can also be used in compilers for register allocation, by applying it to a graph whose vertices represent values to be assigned to registers and whose edges represent conflicts between two values that cannot be assigned to the same register.^[30] In many cases, these interference graphs are chordal graphs, allowing greedy coloring to produce an optimal register assignment.^[31]

In combinatorial game theory, for an impartial game given in explicit form as a directed acyclic graph whose vertices represent game positions and whose edges represent valid moves from one position to another, the greedy coloring algorithm (using the reverse of a topological ordering of the graph) calculates the nim-value of each position. These values can be used to determine optimal play in any single game or any disjunctive sum of games.^[32]

For a graph of maximum degree Δ , any greedy coloring will use at most $\Delta + 1$ colors. Brooks' theorem states that with two exceptions (cliques and odd cycles) at most Δ colors are needed. One proof of Brooks' theorem involves finding a vertex ordering in which the first two vertices are adjacent to the final vertex but not adjacent to each other, and each vertex other than the last one has at least one later neighbor. For an ordering with this property, the greedy coloring algorithm uses at most Δ colors.^[33]

Notes

1. Mitchem (1976).
2. Hoàng & Sritharan (2016), Theorem 28.33, p. 738 (https://books.google.com/books?id=H_IYCwAAQBAJ&pg=PA738); Husfeldt (2015), Algorithm G
3. Frieze & McDiarmid (1997).
4. Welsh & Powell (1967).
5. Johnson (1974); Husfeldt (2015).
6. Kučera (1991); Husfeldt (2015).
7. Husfeldt (2015).
8. Maffray (2003).
9. Rose, Lueker & Tarjan (1976).
10. Chvátal (1984); Husfeldt (2015).
11. Middendorf & Pfeiffer (1990).
12. Zaker (2006).
13. Christen & Selkow (1979).
14. Mitchem (1976); Husfeldt (2015).
15. Matula & Beck (1983).
16. Welsh & Powell (1967); Husfeldt (2015).
17. Matula (1968); Szekeres & Wilf (1968).
18. Kosowski & Manuszewski (2004).
19. Markossian, Gasparian & Reed (1996); Maffray (2003).
20. Markossian, Gasparian & Reed (1996).
21. Gräf, Stumpf & Weißenfels (1998).
22. Brélaz (1979); Lévêque & Maffray (2005).
23. Brélaz (1979).
24. Janczewski et al. (2001).
25. Lévêque & Maffray (2005).
26. Irani (1994).
27. Lovász, Saks & Trotter (1989); Vishwanathan (1992).
28. Kierstead & Trotter (1981).
29. Simmons (1982); Erdős et al. (1987).
30. Poletto & Sarkar (1999). Although Poletto and Sarkar describe their register allocation method as not being based on graph coloring, it appears to be the same as greedy coloring.
31. Pereira & Palsberg (2005).
32. E.g., see Section 1.1 of Nivasch (2006).
33. Lovász (1975).

References

- Brélaz, Daniel (April 1979), "New methods to color the vertices of a graph", *Communications of*

the ACM, **22** (4): 251–256, doi:10.1145/359094.359101 (<https://doi.org/10.1145%2F359094.359101>)

- Christen, Claude A.; Selkow, Stanley M. (1979), "Some perfect coloring properties of graphs", *Journal of Combinatorial Theory, Series B*, **27** (1): 49–59, doi:10.1016/0095-8956(79)90067-4 (<https://doi.org/10.1016%2F0095-8956%2879%2990067-4>), MR 0539075 (<https://www.ams.org/mathscinet-getitem?mr=0539075>)
- Chvátal, Václav (1984), "Perfectly orderable graphs", in Berge, Claude; Chvátal, Václav (eds.), *Topics in Perfect Graphs*, Annals of Discrete Mathematics, **21**, Amsterdam: North-Holland, pp. 63–68. As cited by Maffray (2003).
- Erdős, P.; Hare, W. R.; Hedetniemi, S. T.; Laskar, R. (1987), "On the equality of the Grundy and chromatic numbers of a graph" (https://users.renyi.hu/~p_erdos/1987-18.pdf) (PDF), *Journal of Graph Theory*, **11** (2): 157–159, doi:10.1002/jgt.3190110205 (<https://doi.org/10.1002%2Fjgt.3190110205>), MR 0889347 (<https://www.ams.org/mathscinet-getitem?mr=0889347>).
- Frieze, Alan; McDiarmid, Colin (1997), "Algorithmic theory of random graphs", *Random Structures & Algorithms*, **10** (1–2): 5–42, doi:10.1002/(SICI)1098-2418(199701/03)10:1/2<5::AID-RSA2>3.3.CO;2-6 (<https://doi.org/10.1002%2F%28SICI%291098-2418%28199701%2F03%2910%3A1%2F2%3C5%3A%3AAID-RSA2%3E3.3.CO%3B2-6>), MR 1611517 (<https://www.ams.org/mathscinet-getitem?mr=1611517>).
- Gräf, A.; Stumpf, M.; Weißenfels, G. (1998), "On coloring unit disk graphs", *Algorithmica*, **20** (3): 277–293, doi:10.1007/PL00009196 (<https://doi.org/10.1007%2FPL00009196>), MR 1489033 (<https://www.ams.org/mathscinet-getitem?mr=1489033>).
- Hoàng, Chinh T.; Sritharan, R. (2016), "Chapter 28. Perfect Graphs", in Thulasiraman, Krishnaiyan; Arumugam, Subramanian; Brandstädt, Andreas; Nishizeki, Takao (eds.), *Handbook of Graph Theory, Combinatorial Optimization, and Algorithms*, Chapman & Hall/CRC Computer and Information Science Series, **34**, CRC Press, pp. 707–750, ISBN 9781420011074.
- Husfeldt, Thore (2015), "Graph colouring algorithms", in Beineke, Lowell W.; Wilson, Robin J. (eds.), *Topics in Chromatic Graph Theory*, Encyclopedia of Mathematics and its Applications, **156**, Cambridge University Press, pp. 277–303, arXiv:1505.05825 (<https://arxiv.org/abs/1505.05825>), MR 3380176 (<https://www.ams.org/mathscinet-getitem?mr=3380176>)
- Irani, Sandy (1994), "Coloring inductive graphs on-line", *Algorithmica*, **11** (1): 53–72, doi:10.1007/BF01294263 (<https://doi.org/10.1007%2FBF01294263>), MR 1247988 (<https://www.ams.org/mathscinet-getitem?mr=1247988>).
- Janczewski, R.; Kubale, M.; Manuszewski, K.; Piwakowski, K. (2001), "The smallest hard-to-color graph for algorithm DSATUR", *Discrete Mathematics*, **236** (1–3): 151–165, doi:10.1016/S0012-365X(00)00439-8 (<https://doi.org/10.1016%2FS0012-365X%2800%2900439-8>), MR 1830607 (<https://www.ams.org/mathscinet-getitem?mr=1830607>).
- Kierstead, H. A.; Trotter, W. T. (1981), "An extremal problem in recursive combinatorics", Proceedings of the Twelfth Southeastern Conference on Combinatorics, Graph Theory and Computing, Vol. II (Baton Rouge, La., 1981), *Congressus Numerantium*, **33**: 143–153, MR 0681909 (<https://www.ams.org/mathscinet-getitem?mr=0681909>). As cited by Irani (1994).
- Kosowski, Adrian; Manuszewski, Krzysztof (2004), "Classical coloring of graphs", in Kubale, Marek (ed.), *Graph Colorings*, Contemporary Mathematics, **352**, Providence, Rhode Island: American Mathematical Society, pp. 1–19, doi:10.1090/conm/352/06369 (<https://doi.org/10.1090%2Fconm%2F352%2F06369>), MR 2076987 (<https://www.ams.org/mathscinet-getitem?mr=2076987>)
- Kučera, Luděk (1991), "The greedy coloring is a bad probabilistic algorithm", *Journal of Algorithms*, **12** (4): 674–684, doi:10.1016/0196-6774(91)90040-6 (<https://doi.org/10.1016%2F0196-6774%2891%2990040-6>), MR 1130323 (<https://www.ams.org/mathscinet-getitem?mr=1130323>).

- Johnson, David S. (1974), "Worst case behavior of graph coloring algorithms", *Proceedings of the Fifth Southeastern Conference on Combinatorics, Graph Theory and Computing (Florida Atlantic Univ., Boca Raton, Fla., 1974)*, Congressus Numerantium, **X**, Winnipeg, Manitoba: Utilitas Math., pp. 513–527, MR 0389644 (<https://www.ams.org/mathscinet-getitem?mr=0389644>).
- Lévêque, Benjamin; Maffray, Frédéric (October 2005), "Coloring Meyniel graphs in linear time" (<https://hal.archives-ouvertes.fr/file/index/docid/30179/filename/ColMeyniel.pdf>) (PDF), in Raspaud, André; Delmas, Olivier (eds.), *7th International Colloquium on Graph Theory (ICGT '05), 12–16 September 2005, Hyeres, France*, Electronic Notes in Discrete Mathematics, **22**, Elsevier, pp. 25–28, arXiv:cs/0405059 (<https://arxiv.org/abs/cs/0405059>), doi:10.1016/j.endm.2005.06.005 (<https://doi.org/10.1016%2Fj.endm.2005.06.005>). See also Lévêque, Benjamin; Maffray, Frédéric (January 9, 2006), *Erratum : MCColor is not optimal on Meyniel graphs*, arXiv:cs/0405059 (<https://arxiv.org/abs/cs/0405059>).
- Lovász, L. (1975), "Three short proofs in graph theory", *Journal of Combinatorial Theory, Series B*, **19** (3): 269–271, doi:10.1016/0095-8956(75)90089-1 (<https://doi.org/10.1016%2F0095-8956%2875%2990089-1>), MR 0396344 (<https://www.ams.org/mathscinet-getitem?mr=0396344>).
- Lovász, L.; Saks, M. E.; Trotter, W. T. (1989), "An on-line graph coloring algorithm with sublinear performance ratio", *Discrete Mathematics*, **75** (1–3): 319–325, doi:10.1016/0012-365X(89)90096-4 (<https://doi.org/10.1016%2F0012-365X%2889%2990096-4>), MR 1001404 (<https://www.ams.org/mathscinet-getitem?mr=1001404>).
- Maffray, Frédéric (2003), "On the coloration of perfect graphs", in Reed, Bruce A.; Sales, Cláudia L. (eds.), *Recent Advances in Algorithms and Combinatorics*, CMS Books in Mathematics, **11**, Springer-Verlag, pp. 65–84, doi:10.1007/0-387-22444-0_3 (https://doi.org/10.1007%2F0-387-22444-0_3), ISBN 0-387-95434-1, MR 1952983 (<https://www.ams.org/mathscinet-getitem?mr=1952983>).
- Markossian, S. E.; Gasparian, G. S.; Reed, B. A. (1996), " β -perfect graphs", *Journal of Combinatorial Theory, Series B*, **67** (1): 1–11, doi:10.1006/jctb.1996.0030 (<https://doi.org/10.1006%2Fjctb.1996.0030>), MR 1385380 (<https://www.ams.org/mathscinet-getitem?mr=1385380>).
- Matula, David W. (1968), "A min-max theorem for graphs with application to graph coloring", SIAM 1968 National Meeting, *SIAM Review*, **10** (4): 481–482, doi:10.1137/1010115 (<https://doi.org/10.1137%2F1010115>).
- Matula, David W.; Beck, L. L. (1983), "Smallest-last ordering and clustering and graph coloring algorithms", *Journal of the ACM*, **30** (3): 417–427, doi:10.1145/2402.322385 (<https://doi.org/10.1145%2F2402.322385>), MR 0709826 (<https://www.ams.org/mathscinet-getitem?mr=0709826>).
- Middendorf, Matthias; Pfeiffer, Frank (1990), "On the complexity of recognizing perfectly orderable graphs", *Discrete Mathematics*, **80** (3): 327–333, doi:10.1016/0012-365X(90)90251-C (<https://doi.org/10.1016%2F0012-365X%2890%2990251-C>), MR 1049253 (<https://www.ams.org/mathscinet-getitem?mr=1049253>).
- Mitchem, John (1976), "On various algorithms for estimating the chromatic number of a graph", *The Computer Journal*, **19** (2): 182–183, doi:10.1093/comjnl/19.2.182 (<https://doi.org/10.1093%2Fcomjnl%2F19.2.182>), MR 0437376 (<https://www.ams.org/mathscinet-getitem?mr=0437376>).
- Nivasch, Gabriel (2006), "The Sprague–Grundy function of the game Euclid", *Discrete Mathematics*, **306** (21): 2798–2800, doi:10.1016/j.disc.2006.04.020 (<https://doi.org/10.1016%2Fj.disc.2006.04.020>), MR 2264378 (<https://www.ams.org/mathscinet-getitem?mr=2264378>).
- Pereira, Fernando Magno Quintão; Palsberg, Jens (2005), "Register allocation via coloring of chordal graphs", in Yi, Kwangkeun (ed.), *Programming Languages and Systems: Third Asian Symposium, APLAS 2005, Tsukuba, Japan, November 2–5, 2005, Proceedings*, Lecture Notes in Computer Science, **3780**, Springer, pp. 315–329, doi:10.1007/11575467_21 (https://doi.org/10.1007%2F11575467_21).
- Poletto, Massimiliano; Sarkar, Vivek (September 1999), "Linear scan register allocation", *ACM Transactions on Programming Languages and Systems*, **21** (5): 895–913, doi:10.1145/330249.330250 (<https://doi.org/10.1145%2F330249.330250>).

- Rose, D.; Lueker, George; Tarjan, Robert E. (1976), "Algorithmic aspects of vertex elimination on graphs", *SIAM Journal on Computing*, **5** (2): 266–283, doi:10.1137/0205021 (<https://doi.org/10.1137/0205021>), MR 0408312 (<https://www.ams.org/mathscinet-getitem?mr=0408312>).
- Simmons, Gustavus J. (1982), "The ordered chromatic number of planar maps", Proceedings of the thirteenth Southeastern conference on combinatorics, graph theory and computing (Boca Raton, Fla., 1982), *Congressus Numerantium*, **36**: 59–67, MR 0726050 (<https://www.ams.org/mathscinet-getitem?mr=0726050>)
- Sysło, Maciej M. (1989), "Sequential coloring versus Welsh–Powell bound", *Discrete Mathematics*, **74** (1–2): 241–243, doi:10.1016/0012-365X(89)90212-4 ([https://doi.org/10.1016/0012-365X\(89\)90212-4](https://doi.org/10.1016/0012-365X(89)90212-4)), MR 0989136 (<https://www.ams.org/mathscinet-getitem?mr=0989136>).
- Szekeres, George; Wilf, Herbert S. (1968), "An inequality for the chromatic number of a graph", *Journal of Combinatorial Theory*, **4**: 1–3, doi:10.1016/S0021-9800(68)80081-X ([https://doi.org/10.1016/S0021-9800\(68\)80081-X](https://doi.org/10.1016/S0021-9800(68)80081-X)).
- Vishwanathan, Sundar (1992), "Randomized online graph coloring", *Journal of Algorithms*, **13** (4): 657–669, doi:10.1016/0196-6774(92)90061-G ([https://doi.org/10.1016/0196-6774\(92\)90061-G](https://doi.org/10.1016/0196-6774(92)90061-G)), MR 1187207 (<https://www.ams.org/mathscinet-getitem?mr=1187207>).
- Welsh, D. J. A.; Powell, M. B. (1967), "An upper bound for the chromatic number of a graph and its application to timetabling problems", *The Computer Journal*, **10** (1): 85–86, doi:10.1093/comjnl/10.1.85 (<https://doi.org/10.1093/comjnl/10.1.85>).
- Zaker, Manouchehr (2006), "Results on the Grundy chromatic number of graphs", *Discrete Mathematics*, **306** (2–3): 3166–3173, doi:10.1016/j.disc.2005.06.044 (<https://doi.org/10.1016/j.disc.2005.06.044>), MR 2273147 (<https://www.ams.org/mathscinet-getitem?mr=2273147>).

Retrieved from "https://en.wikipedia.org/w/index.php?title=Greedy_coloring&oldid=1024512218"

This page was last edited on 22 May 2021, at 15:45 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.