CSE6140 Final Project

AVERY BODENSTEIN*, Georgia Institute of Technology, USA
ADRIAN THINNYUN*, Georgia Institute of Technology, USA
JAI JACOB*, Georgia Institute of Technology, USA
ZHEYI ZHANG*, Georgia Institute of Technology, USA

A clear and well-documented LaTeX document is presented as an article formatted for publication by ACM in a conference proceedings or journal publication. Based on the "acmart" document class, this article presents and explains many of the common variations, as well as many of the formatting elements an author may use in the preparation of the documentation of their work.

CCS Concepts: • **Theory of computation** → **Graph algorithms analysis**; **Branch-and-bound**; **Approximation algorithms analysis**.

Additional Key Words and Phrases: minimum vertex cover, spanning, topology, heuristics

ACM Reference Format:
Avery Bodenstein, Adrian Thinnyun, Jai Jacob, and Zheyi Zhang. 2022. CSE6140 Final Project. *Proc. ACM Meas. Anal. Comput. Syst.* 37, 4, Article 111 (August 2022), 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

short summary of the problem, the approach and the results you have obtained

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam

*All authors contributed equally to this research.

Authors' addresses: Avery Bodenstein, abodenstein3@gatech.edu, Georgia Institute of Technology, North Ave NW, Atlanta, Georgia, USA, 30332; Adrian Thinnyun, Georgia Institute of Technology, North Ave NW, Atlanta, Georgia, USA, 30332; Jai Jacob, Georgia Institute of Technology, North Ave NW, Atlanta, Georgia, USA, 30332; Zheyi Zhang, Georgia Institute of Technology, North Ave NW, Atlanta, Georgia, USA, 30332.

2476-1249/2022/8-ART111 $15.00
https://doi.org/XXXXXXX.XXXXXXX

cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

2 PROBLEM DEFINITION

formal definition of the problem

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

3 RELATED WORK

a short survey of existing work on the same problem, and important results in theory and practice

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

4 ALGORITHMS

Detailed description of each algorithm you have implemented, with pseudo-code, approx- imation guarantee (if any), time and space complexities, etc. What are the potential strengths and weaknesses of each type of approach? Did you use any kind of automated tuning or configuration for your local search? Why and how you chose your local search approaches and their components? Please cite any sources of information that you used to inform your algorithm design

4.1 Branch and Bound

4.1.1 Description. The branch and bound approach is a non-polynomial time algorithm which is guaranteed to return an optimal solution. The BnB algorithm at worst does not perform any better than a brute force search, but can have significant performance gains where portions of the search space can be pruned. This pruning is performed by iteratively defining sub-problems and determining lower bounds on the cost to go for these sub-problems. If the best possible cost for a sub-problem exceeds the current best cost that subproblem (and therefore all it's derivatives) can be pruned from the search space. The sub-problems presented in this algorithm are defined based on inclusion or exclusion of nodes from the cover set.

4.1.2 Pseudo Code. -

The overall branch and bound algorithm has the following structure:

```
Data: P
F ← (∅, P);
B ← (+∞, (∅, P));
while F not empty do
    choose (X, Y) in F;
    expand (X, Y);
    let (x_1, y_1), (x_2, y_2) be new configurations;
    foreach (x_i, y_i) do
        if solution found then
            if cost(x_i) < B then
                B ← (cost(x_i), (x_i, y_i));
            end
        end
        if not dead end then
            if lowerBound(x_i) < B then
                F ← F ∪ (x_i, y_i);
            end
        end
    end
end
return (B)
```

Algorithm 1: Branch and Bound

where P is the graph, x_i are all the vertices in the cover set, and y_i are the vertices remaining for selection. There are four sub functions, "choose", "expand", "checkSolution", "checkDeadEnd", and "lowerBound". In the implementation presented here, choose selects the subproblem in F with the lowest lowerBound on cost to go. Expand takes that subproblem and returns two subproblems, one with the node with fewest unique edges selected and one without that node as a possible selection. checkSolution checks to see if all edges are covered by x_i. checkDeadEnd checks if $x_i \cup y_i$ covers all edges. Finally lowerBound runs the heuristic algorithm described in this report on y_i then divides by the approximation ratio (2) to get a lower bound on the possible additional nodes required to cover all edges. This is then added to the number of nodes already in x_i.

4.1.3 Algorithm Analysis. Additional description

Gives exact solution

Time and space complexity

4.2 Construction Heuristic

4.2.1 Description. The construction heuristic approach implemented here is the *Greedy Independent Cover (GIC)* algorithm, presented in Halldórsson and Radhakrishnan [1994] [3] for the independent set problem. This is an approximation approach which runs in polynomial time and yields an answer within a bound of the optimal solution. Runtime and Approximation Ratio analysis is provided in 4.2.3. This algorithm lends itself well to use with priority queues. All vertices in the graph are stored in a priority queue, ordered by the number of adjacent edges remaining in G: q[nodeLabel] = [nEdges,destinationNodes]. At each iteration the vertex with the fewest remaining edges is popped from the queue. Each of it's neighbors is then removed from the queue and added to the cover set. For each remaining edge in each neighbor, the source node is removed from the list of edges in the destination node and the number of edges in the destination node is lowered by one. This process continues until all edges are covered.

4.2.2 Pseudo Code. -

```
Data: G = (V,E)
C ← ∅;
while E ≠ ∅ do
    select a vertex u of minimum degree;
    C ← C ∪ N(u);
    V ← V - (N(u) ∪ u);
end
return C;
```

Algorithm 2: Greedy Independent Cover (GIC)

Where N(u) is the neighborhood (all adjacent vertices to) u.

```
Data: q,nEdges
C ← 0;
while C < nEdges do
    u ← q.peek();
    foreach neigbor i in N(u) do
        C ← C + q[i][0];
        foreach edge (i,j) in neighbor do
            q[j][0] ← q[j][0] - 1;
            q[j][1] ← q[j][1] - {i};
        end
    end
    q = q - u;
end
return C;
```

Algorithm 3: Detailed Implementation

4.2.3 Algorithm Analysis. -

As discussed in Delbot and Laforest [2010] [2] the Greedy Independent Cover algorithm performs extremely well. As can be seen in table 1 the relative error for GIC never exceeds 6% (and this is on the Jazz graph where the estimated solution is only 1 node larger than the optimal solution). However, the disadvantage of GIC is that the worst case performance is relatively poor. As shown in Avis and Imamura [2007] [1], the approximation ratio is at least $\frac{\sqrt{\Delta}}{2}$ where Δ is the maximum degree in G.

Figure 1 shows a comparison between the approximation ratios of several common algorithms described in [2]. Note that while GIC performs quite well, it's approximation ratio exceeds Depth First Search, or either of the Edge Deletion bounds for any Δ above 16.

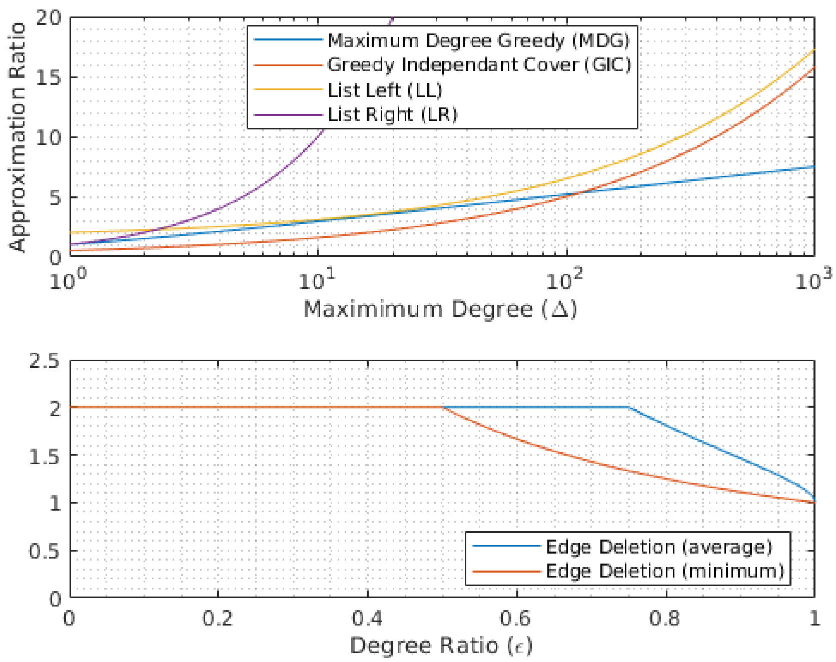

Fig. 1. Comparison of approximation ratios between several Vertex Cover heuristics

The time complexity of the GIC algorithm as implemented is $O(n^2)$ where n is the number of nodes. As each node is selected via the minimum cardinality, at worst it has zero neighbors, so there would be at worst n node selections. However in this case there would be 0 edge updates. For every neighbor that is present there are edge updates equal to the neighbors' cardinality. This can be at most $n - i$, where i is the number of previous node selections. The complete complexity is then $O(\frac{n}{2}(n+1))$.

4.3 Local Search 1 (specific descriptor)

4.3.1 Description. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

4.3.2 Pseudo Code.

```
Data: B,C
n_a = 0;
i = len(B);
j = len(C);
A = [0]*(i+j-2);
for l = i+j-2; l ≥ 0; l-=1 do
    if i == 0 then
        A[0:l] = C[0:j];
        return (A,n_a)
    else if j == 0 then
        A[0:l] = B[0:i];
        return (A,n_a)
    if C[j-1] ≥ B[i-1] then
        A[l] = C[j-1];
        j = j-1;
    else
        A[l] = B[i-1];
        n_a = n_a + j;
        i = i-1;
end
return (A,n_a)
```

Algorithm 4: merge

4.3.3 Algorithm Analysis. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

4.4 Local Search 2 (specific descriptor)

4.4.1 Description. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

4.4.2 Pseudo Code.

```
Data: B,C
n_a = 0;
i = len(B);
j = len(C);
A = [0]*(i+j-2);
for l = i+j-2; l ≥ 0; l-=1 do
    if i == 0 then
        A[0:l] = C[0:j];
        return (A,n_a)
    else if j == 0 then
        A[0:l] = B[0:i];
        return (A,n_a)
    if C[j-1] ≥ B[i-1] then
        A[l] = C[j-1];
        j = j-1;
    else
        A[l] = B[i-1];
        n_a = n_a + j;
        i = i-1;
end
return (A,n_a)
```

Algorithm 5: merge

4.4.3 Algorithm Analysis. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

5 EMPIRICAL EVALUATION

a detailed description of your platform (CPU, RAM, language, compiler, etc.), experimental procedure, evaluation criteria and obtained results (plots, tables, etc.). What is the lower bound on the optimal solution quality that you can drive from the results of your approximation algorithm and how far is it from the true optimum? How about from your branch-and-bound?

Table 1. Algorithm Performance

	Branch and Bound			Construction Heuristic			Local Search 1			Local	
Dataset	Time(s)	VC Value	RelErr	Time(s)	VC Value	RelErr	Time(s)	VC Value	RelErr	Time(s)	V
jazz				0.0037	159	0.063					
karate				0.0005	14	0.00					
football				0.0014	95	0.011					
as-22july06				0.21	3303	0.00					
hep-th				0.098	3943	0.0043					
star				0.24	7069	0.024					
star2				0.21	4674	0.029					
netscience				0.017	901	0.0022					
email				0.015	604	0.017					
delaunay n10				0.012	733	0.043					
power				0.051	2226	0.010					

6 DISCUSSION

a comparative analysis of how different algorithms perform with respect to your evaluation criteria, or expected time complexity, etc

7 CONCLUSION

8 MATH EQUATIONS

You may want to display math equations in three distinct styles: inline, numbered or non-numbered display. Each of the three are discussed in the next sections.

8.1 Inline (In-text) Equations

A formula that appears in the running text is called an inline or in-text formula. It is produced by the **math** environment, which can be invoked with the usual `\begin ...\end` construction or with the short form `$ ...$`. You can use any of the symbols and structures, from α to ω, available in LaTeX [?]; this section will simply show a few examples of in-text equations in context. Notice how this equation: $\lim_{n\rightarrow \infty} x = 0$, set here in in-line math style, looks slightly different when set in display style. (See next section).

8.2 Display Equations

A numbered display equation—one set off by vertical space from the text and centered horizontally—is produced by the **equation** environment. An unnumbered display equation is produced by the **displaymath** environment.

Again, in either environment, you can use any of the symbols and structures available in LaTeX; this section will just give a couple of examples of display equations in context. First, consider the equation, shown as an inline equation above:

$$\lim_{n\rightarrow \infty} x = 0 \tag{1}$$

Notice how it is formatted somewhat differently in the **displaymath** environment. Now, we'll enter an unnumbered equation:

$$\sum_{i=0}^{\infty} x + 1$$

and follow it with another numbered equation:

$$\sum_{i=0}^{\infty} x_i = \int_0^{\pi+2} f \tag{2}$$

just to demonstrate LaTeX's able handling of numbering.

9 FIGURES

The "`figure`" environment should be used for figures. One or more images can be placed within a figure. If your figure contains third-party material, you must clearly identify it as such, as shown in the example below.

Your figures should contain a caption which describes the figure to the reader.

Figure captions are placed *below* the figure.

Every figure should also have a figure description unless it is purely decorative. These descriptions convey what's in the image to someone who cannot see it. They are also used by search engine crawlers for indexing images, and when images cannot be loaded.

A figure description must be unformatted plain text less than 2000 characters long (including spaces). **Figure descriptions should not repeat the figure caption – their purpose is to capture important information that is not already provided in the caption or the main text of the paper.** For figures that convey important and complex new information, a short text description may not be adequate. More complex alternative descriptions can be placed in an appendix and referenced in a short figure description. For example, provide a data table capturing the information in a bar chart, or a structured list representing a graph. For additional information regarding how best to write figure descriptions and why doing this is so important, please see https://www.acm.org/publications/taps/describing-figures/.

10 CITATIONS AND BIBLIOGRAPHIES

The use of BibTeX for the preparation and formatting of one's references is strongly recommended. Authors' names should be complete — use full first names ("Donald E. Knuth") not initials ("D. E. Knuth") — and the salient identifying features of a reference should be included: title, year, volume, number, pages, article DOI, etc.

The bibliography is included in your source document with these two commands, placed just before the `\end{document}` command:

```
\bibliographystyle{ACM-Reference-Format}
\bibliography{bibfile}
```

where "`bibfile`" is the name, without the "`.bib`" suffix, of the BibTeX file.

Citations and references are numbered by default. A small number of ACM publications have citations and references formatted in the "author year" style; for these exceptions, please include this command in the **preamble** (before the command "`\begin{document}`") of your LaTeX source:

```
\citestyle{acmauthoryear}
```

Some examples. A paginated journal article [2] [1] [3]

Fig. 2. 1907 Franklin Model D roadster. Photograph by Harris & Ewing, Inc. [Public domain], via Wikimedia Commons. (https://goo.gl/VLCRBB).

11 ACKNOWLEDGMENTS

Identification of funding sources and other support, and thanks to individuals and groups that assisted in the research and the preparation of the work should be included in an acknowledgment section, which is placed just before the reference section in your document.

This section has a special environment:

```
\begin{acks}
...
\end{acks}
```

so that the information contained therein can be more easily collected during the article metadata extraction phase, and to ensure consistency in the spelling of the section heading.

Authors should not prepare this section as a numbered or unnumbered `\section`; please use the "`acks`" environment.

ACKNOWLEDGMENTS

To Robert, for the bagels and explaining CMYK and color spaces.

REFERENCES

[1] David Avis and Tomokazu Imamura. 2007. A list heuristic for vertex cover. *Operations Research Letters* 35, 2 (2007), 201–204. https://doi.org/10.1016/j.orl.2006.03.014

[2] François Delbot and Christian Laforest. 2010. Analytical and Experimental Comparison of Six Algorithms for the Vertex Cover Problem. *ACM J. Exp. Algorithmics* 15, Article 1.4 (nov 2010), 27 pages. https://doi.org/10.1145/1671970.1865971

[3] Magnús Halldórsson and Jaikumar Radhakrishnan. 1997. Greed is Good: Approximating Independent Sets in Sparse and Bounded-Degree Graphs. *Algorithmica* 18 (05 1997), 145–163. https://doi.org/10.1007/BF02523693

Received 4 December 2022