NCUR 2021 Proceedings

Automatic Crossword Generation

Computer Science - Time: Tue 12:30pm-1:30pm - Session Number: 4043
Otis Peterson, Michael Wehar, Department of Computer Science, Swarthmore College, 500 College Ave, Swarthmore, PA 19081
Otis Peterson

Algorithms Design is a field of study within computer science that is concerned with finding efficient algorithms that solve computational problems. There are many computational problems that we still do not know how to solve efficiently such as NP-hard problems. For our research project, we investigated computationally hard problems related to two dimensional word puzzles. In particular, we focused on the Crossword Construction Problem which is known to be NP-complete. Given a grid of shaded/empty cells and a wordlist, the Crossword Construction Problem asks if there exists a fill for the grid with letters such that every contiguous block of empty cells, horizontally and vertically, forms a word from the wordlist. In short, given a blank crossword grid, we are asking how to fill in every empty cell such that every across and down is a valid word. Crossword puzzles are regularly published in many leading newspapers and media outlets with millions of avid crossword puzzlers. As a result, Crossword puzzles are a well established puzzle with many common formats and rules. Although there are tools that assist with crossword puzzle construction, many crossword puzzle publishers still create crosswords by hand. Crossword puzzle publishers rely on submissions because there is still no accessible construction tool that is efficient and effective enough for publishers. For our project, we analyzed existing approaches for solving the crossword construction problem and implemented our own heuristic algorithms (based on established approaches) as part of a web-based crossword construction application. Although our algorithms theoretically run in exponential time, they are surprisingly efficient on real world examples. Going forward, we hope to make our crossword construction application publicly available, develop more performance metrics to compare our crossword construction algorithm with existing tools, and explore new algorithms to advance the field of automatic crossword puzzle construction.

Algorithms Design is a field of study within computer science that is concerned with finding efficient algorithms that solve computational problems. There are many computational problems that we still do not know how to solve efficiently such as NP-hard problems. For our research project, we investigated computationally hard problems related to two dimensional word puzzles. In particular, we focused on the Crossword Construction Problem which is known to be NP-complete. Given a grid of shaded/empty cells and a wordlist, the Crossword Construction Problem asks if there exists a fill for the grid with letters such that every contiguous block of empty cells, horizontally and vertically, forms a word from the wordlist. In short, given a blank crossword grid, we are asking how to fill in every empty cell such that every across and down is a valid word. Crossword puzzles are regularly published in many leading newspapers and media outlets with millions of avid crossword puzzlers. As a result, Crossword puzzles are a well established puzzle with many common formats and rules. Although there are tools that assist with crossword puzzle construction, many crossword puzzle publishers still create crosswords by hand. Crossword puzzle publishers rely on submissions because there is still no accessible construction tool that is efficient and effective enough for publishers. For our project, we analyzed existing approaches for

solving the crossword construction problem and implemented our own heuristic algorithms (based on established approaches) as part of a web-based crossword construction application. Although our algorithms theoretically run in exponential time, they are surprisingly efficient on real world examples. Going forward, we hope to make our crossword construction application publicly available, develop more performance metrics to compare our crossword construction algorithm with existing tools, and explore new algorithms to advance the field of automatic crossword puzzle construction.

Efficient Algorithms for the Four Corners Problem and Two-Dimensional Pattern Matching

Computer Science - Time: Tue 12:30pm-1:30pm - Session Number: 4042
Ari Liloia and Michael Wehar, Department of Computer Science, Swarthmore College, 500 College Avenue, Swarthmore PA 19081
Ari Liloia

Two dimensional pattern matching is concerned with the recognition of patterns within two dimensional data structures, such as binary matrices, which are matrices whose elements have one of two possible values. There are many algorithmic problems related to efficiently recognizing patterns within binary matrices. One such problem asks: Given a rectangular matrix made up of ones and zeros, does there exist a submatrix whose four corners are all ones? This problem is referred to as the Four Corners Problem. It appears on many algorithms challenge websites and it has applications to the well known frequent itemsets problem from data mining. The standard solution for the Four Corners Problem runs in cubic time. Through an ongoing research project with my advisor (and several collaborators) we were able to improve the standard solution by designing and implementing a more efficient algorithm. In particular, we were able to introduce nearly quadratic time algorithms for all common variations of the Four Corners Problem over a binary alphabet, each of which involves one of the four possible configurations of corner values (1111, 1011, 1001, or 1010). We then implemented our algorithms in Python and made our code available in an open source repository on GitHub, so that other researchers can use these algorithms to solve new problems. Our code was written following principles of software design that advocate for simplicity and readability, making it ideal for usage in a classroom setting for educational purposes. In addition, we showed that the Four Corners Problem is equivalent to several problems related to data mining and computational geometry, offering applications of our work to other problems within computer science, including frequent itemset mining, the buyer-seller matching problem, and finding rectangles in sets of points.

Two dimensional pattern matching is concerned with the recognition of patterns within two dimensional data structures, such as binary matrices, which are matrices whose elements have one of two possible values. There are many algorithmic problems related to efficiently recognizing patterns within binary matrices. One such problem asks: Given a rectangular matrix made up of ones and zeros, does there exist a submatrix whose four corners are all ones? This problem is referred to as the Four Corners Problem. It appears on many algorithms challenge websites and it has applications to the well known frequent itemsets problem from data mining. The standard solution for the Four Corners Problem runs in cubic time. Through an ongoing research project with my advisor (and several collaborators) we were able to improve the standard solution by designing and implementing a more efficient algorithm. In particular, we were able to introduce nearly quadratic time algorithms for all common variations of the Four Corners Problem over a binary alphabet, each of which involves one of the four possible configurations of corner values (1111, 1011, 1001, or 1010). We then implemented our algorithms in Python and made our code available in an open source repository on GitHub, so that other researchers

can use these algorithms to solve new problems. Our code was written following principles of software design that advocate for simplicity and readability, making it ideal for usage in a classroom setting for educational purposes. In addition, we showed that the Four Corners Problem is equivalent to several problems related to data mining and computational geometry, offering applications of our work to other problems within computer science, including frequent itemset mining, the buyer-seller matching problem, and finding rectangles in sets of points.

Machine Assisted Speed Reading and Training

Computer Science - Time: Wed 1:30pm-2:30pm - Session Number: 1013
Christopher Brandt, Michael Wehar, Computer Science, Swarthmore College, 500 College Avenue, Swarthmore PA, 19081
Christopher Brandt

Educational technology (also known as EdTech) is a developing field that combines computer software and educational theory. Within EdTech we explored technology for reading. Reading is a vast subject that combines elements of human anatomy and psychology. For our project, we explore how technology can lead to more consistent eye movement, faster information processing, and comprehension to help people read more efficiently. In the United States, reading is typically only taught through elementary school, so people miss the opportunity to continue advancing their reading skills throughout their lives. Our hope is to use technology to enable users to continue advancing their reading skills.

Based on our findings, we propose a web-based system called RhinoReading to help improve reading speed through visual effects. Our system is a Web Application written using HTML, JavaScript, and CSS. It uses visual effects to guide eye movement across a virtual reading page. The reading page includes visual configurations for color and font, tracing features such as highlight, underline, and bold, speed selection by word or letter, and an assortment of tracking options that serve a variety of purposes.

The second part of this project involves server management and data analysis. Our system's backend was implemented as a Flask Application programmed in Python. This app controls flow between pages and processes data transfer between pages and a relational database. The pages include home, reading passage selection, comprehension quizzes, and statistics. The database stores user accounts, passages, quiz questions/answers, and reading session data.

By making our system available to the public, we hope to gain feedback from beta users and learn about the effectiveness of technology for the improvement of reading speed.

Educational technology (also known as EdTech) is a developing field that combines computer software and educational theory. Within EdTech we explored technology for reading. Reading is a vast subject that combines elements of human anatomy and psychology. For our project, we explore how technology can lead to more consistent eye movement, faster information processing, and comprehension to help people read more efficiently. In the United States, reading is typically only taught through elementary school, so people miss the opportunity to continue advancing their reading skills throughout their lives. Our hope is to use technology to enable users to continue advancing their reading skills.

Based on our findings, we propose a web-based system called RhinoReading to help improve reading speed through visual effects. Our system is a Web Application written using HTML, JavaScript, and

CSS. It uses visual effects to guide eye movement across a virtual reading page. The reading page includes visual configurations for color and font, tracing features such as highlight, underline, and bold, speed selection by word or letter, and an assortment of tracking options that serve a variety of purposes.

The second part of this project involves server management and data analysis. Our system's backend was implemented as a Flask Application programmed in Python. This app controls flow between pages and processes data transfer between pages and a relational database. The pages include home, reading passage selection, comprehension quizzes, and statistics. The database stores user accounts, passages, quiz questions/answers, and reading session data.

By making our system available to the public, we hope to gain feedback from beta users and learn about the effectiveness of technology for the improvement of reading speed.
