Assessment of Introducing Algorithms with Video Lectures and Pseudocode Rhymed to a Melody

*Anonymous*

**ABSTRACT**

Can videos with audiovisual demonstrations and music be used to introduce computing concepts so that students cultivate conceptual grasp and increased self-confidence?  In this paper, we discuss two video series which serve as an introduction to two fundamental algorithms: *binary search* and *selection sort*.  Each series begins with an overview of the algorithm with step-by-step examples over an interactive blackboard. It proceeds to a video that illustrates how to perform a complexity analysis with guided examples and then applies that process to the associated algorithm. The series concludes with a video showcasing a song with algorithm pseudocode as lyrics, which are utilized line by line to compose the algorithm in code. These video series were piloted among a set of introductory courses involving coding and algorithmic concepts at two colleges. We assess the effectiveness of each series in terms of conceptual understanding and changes in student attitudes.

**Categories and Subject Descriptors**

K.3.m [**Computers and Education**]: Miscellaneous

**Keywords**

Flipped classroom, videos, music, song, tools for novices.

# INTRODUCTION

For thousands of years, music has been used to help people to remember stories or concepts important enough to be shared and passed down through generations, including not only those of heritage and religion but of academics. Examples include the *Iliad* and the *Odyssey* which are written in hexameter rhythm, [Mitchell], the Old Testament which is composed in verse and chanted with trope tones, and the Mongol Empire military practice of sending secret messages from China to Europe in the form of song with varying tones and trills [mnemonics in the 21st century]. The basis behind the effectiveness of music and rhyme as a mnemonic tool comes from its inherent organizational structure of timing and pitch, which simplifies and expedites the retrieval of complex information [Roediger,189]. Moreover, music serves as a unique bridge between parts of the brain with specific functions, enabling global understanding and in turn flexible application of ideas [Thaut,19 CCSE]. In fact, there are instances where music has been successfully utilized by people with brain trauma such as multiple sclerosis and Alzheimer’s Disease as a tool for retrieval of verbal information [Thaut]. Here, the global nature of the neural encoding of music allowed stronger areas of the brain to compensate for compromised ones in the collection of information. In the classroom, rhythm and song have been demonstrated to support recall for a diverse population such as young, novice students [Yeoh] as well as students with learning disabilities [Claussen]. Particularly, in computer science education, music has been used to understand complexity in courses ranging from introductory [**Dougherty**] to analysis of algorithms [Knuth, Chavey]. For a more extensive overview about the use of songs in learning, especially with computing, see [**Schreiber**].

SAMPLE: Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*SIGCSE’17*, March 8–11, 2017, Seattle, WA, USA.

Copyright 2017 ACM 1-58113-000-0/00/0010 …$15.00.

DOI: http://dx.doi.org/10.1145/12345.67890

*SIGCSE’17*, March 8–11, 2017, Seattle, WA, USA.

Copyright 2017 ACM 1-58113-000-0/00/0010 …$15.00.

DOI: http://dx.doi.org/10.1145/12345.67890

Generally, faculty members have full schedules and demanding curricula. With so much material to cover and so little time it can be difficult to develop lectures that incorporate discussion and problem solving. Additionally, faculty often face the challenge of engaging the entire class and ensuring that all students are following the lectures and not falling behind. Part of what makes this so hard is that lecture is often students’ first exposure to the material which in computer science often entails processing long flows of abstract ideas. The default solution to this problem is assigning textbook reading before class, but this can be unappealing and ineffective for students since the information is static, purely visual, dense, dry and full of notation. To address these challenges we propose a video series that students can view before class that incorporates voice-guided, animated examples and emphasizes intuition and connections to existing knowledge over notation and jargon. This approach can appeal to learners of all types since the information is dynamic, audiovisual, musical, and delivered at the user’s pace.

Specifically, this project outlines…

Thus, any tool or approach that can be administered outside of class meetings and has the potential to increase learning is useful and even desired as in [Maher].  This project outlines such a set of readily available videos that introduce two algorithms: binary search and selection sort.  In each series of three videos, there is a simple lecture covering intuition followed by a video introducing the idea of complexity analysis, and then a song presented with captioned lyrics that serves as both a mnemonic device and pseudocode.  A demonstration shows how these lyrics are then “cut and pasted” directly into a program and used as comments for the associated lines of code they represent.  We expect this approach will be of great value to beginner computing students where such a “musical scaffold” will offer a balance of accessibility, utility, and perhaps a form of entertainment.

We had two primary goals for what students gain from these videos: technical grasp, and enhanced self-confidence about the algorithm. Our technical objective for students was to be able to simulate the algorithm, understand the complexity analysis, and remember important facts about the algorithm. These factors we felt would guide students in being able to code the algorithm and recognize when it should or should not be used.

Although it can easily be overlooked in mathematics, science, and engineering there is an integral *psychological* element in the instruction of these subjects. In particular, the belief in oneself academically is a critical component in an individual’s academic performance. For instance, in a meta-analysis of 109 studies conducted by the American College Test (ACT) academic factors such as high school GPA, ACT score and non-academic factors such as institutional commitment, and financial situation were used to predict college GPA. Of all of the factors on the list academic-self confidence had the strongest correlation with college GPA. Therefore in our study, reduced intimidation and academic self-confidence about the material were core to our assessment.

We first conducted a preliminary investigation in the Fall of 2015, assigning the Binary Search video series in conjunction with two sections of the same CS1 course taught by two different professors at the same college [**Schreiber**].  This paper reports on an expanded investigation in the Spring of 2016.

## Environment of this Investigation

The two video series (available online for free at <website name>) were used among the following three distinct computing courses:

1. A CS1 course at *College A* taught by three different instructors in corresponding sections.
2. A new CS1 course that provided emphasis on programming, including imperative and object-based approaches, that serves as an alternate entry into the CS major at *College B*, taught by a fourth professor distinct from the above three professors.
3. A CS2 course at *College B* taught by the same professor at *College B* above in course #2.

Of the courses described above, courses #1 and #2 used both video series during the course.  Course #3 only viewed Selection Sort for this report as these students had already viewed the Binary Search video series as part of [**Schreiber**].

Each professor was asked to assign, or at least “strongly encourage,” the students in their course to review a given video series just before the same algorithm was introduced in the course.  We had hoped that students reviewed the material prior to the class where the corresponding algorithm was discussed, but we had no way of ensuring each student’s motivation, time management and/or behavior (insert percentages of students from each class who participated).

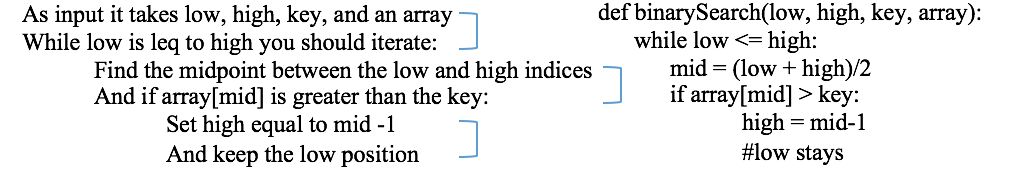


Figure Snippet of Binary Search Song Mapped to Equivalent Python

Before each video series was a request to complete a brief attitude survey (*i.e.,* pre-survey).  Next, each student would complete the three videos in sequence for the given algorithm.  Finally, students completed a second attitude survey (*i.e.,* post-survey) with follow-up questions on the initial survey, as well as a few conceptual questions, and more questions about their experience and takeaways from watching the videos.

## The Video Series

In this study we examine the effectiveness of two video series. Both of the series employ a three-part approach to teaching algorithms: intuition, analysis, and then song and code. The intuition video explains the idea behind the algorithm and how one could perform it using pen and paper– thinking using the same process as a program would upon running the algorithm. The instructor in the video first covers a high-level explanation of the algorithm and then performs it step by step on an interactive blackboard, where the result of each iteration are visible to the viewer. Next, the idea of asymptotic analysis is introduced to the viewer. The instructor accomplishes this by showing how an algorithm can be broken down into a number of steps and then distilling this value by eliminating constants and terms that are dominated as the input size grows. After observing a few simple analysis examples, the viewer witnesses how to synthesize the algorithm into a generic number of steps (in terms of a list of size *n*) and then how to apply the process of distillation they just learned. After covering the analysis, the instructor performs the “*anonymous*” which is the given algorithm’s pseudo code in rhymed couplets paired with melody as a song with piano accompaniment. During this song, the lyrics are visualized in a code editor. Finally, the instructor takes each line of pseudo code and shows how it serves as the structure of an algorithm built in Python code (the language chosen for its simplicity). No prerequisite Python knowledge is assumed, but students can visit the website *anonymous.url* for introductory explanations of Python syntax and concepts.

## Study Methodology

We administered both a pre- and post- survey, each to assess the effectiveness of a single video series used to introduce a particular algorithm. Specifically, students would answer questions that gauged their familiarity with programming, music and the given algorithm itself, and attitudinally, how confident they were with their understanding of the algorithm. In order to track their performance, students provided an anonymous ID (last four digits of phone number + last three digits of zip) on each of the surveys that allowed us to pair each person’s pre responses with their post response. Note that if a student did not provide their ID in both their pre and post surveys we could not consider their responses in the calculation of average improvement.

Since objectives are stated at the very beginning of each video, students were instructed that if they felt that they had a strong grasp of the objective of those video objectives they could move onto the next in the series. The post survey consisted of questions that gauged any change in students’ level of confidence reported after watching the video series, but also assessed each student’s knowledge by asking conceptual questions about the algorithm and its analysis. We offered a section where students could optionally provide their gender, race/ethnicity, and whether they had any disability because we were also interested in measuring responses across different demographics. At the end of the survey, students had the option to write feedback of any nature about the videos. Mention how professor instruction fit into the video series

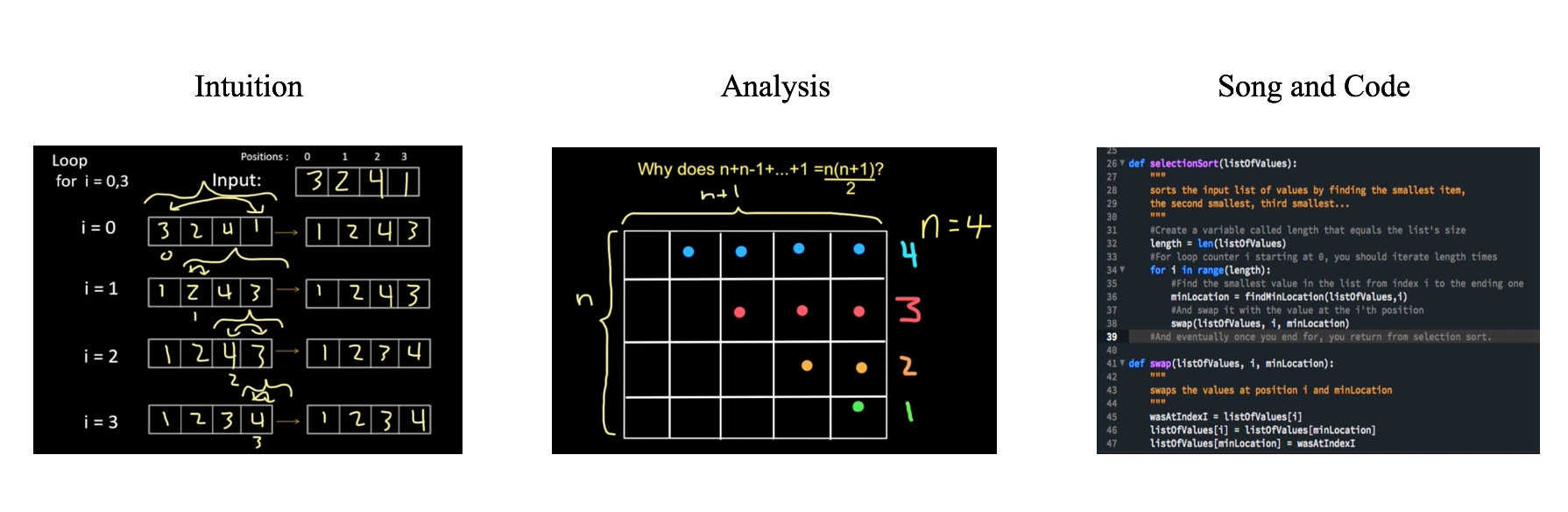


Figure Video Series Sequence

# RESULTS

We first describe the makeup of the students in our study as reported by the students themselves. Next, we present results of the students understanding of the conceptual materials obtained from the post survey. Finally, we try to gauge any change in confidence and attitude reported by the students.

## Demographics

The classes surveyed came from three different courses between two colleges. For both algorithms we requested and recorded the distributions for race/ethnicity, course/section, disability and gender. Students were invited to share each item with an open-ended request, and not from a fixed set of choices.

We present the student reported demographics for the study in the following three tables. In addition to these characteristics we also invited students to share any disability. Of the 22 (out of 35 total) that responded to the disability question for binary search one student reported ADHD, and of the 28 (out of 53 total) students that responded to the disability questions for selection sort, again only one reported having ADHD. **[JD: This paragraph is confusing to me at present]**

The data in Tables 1 – 3 below represent the entire post-survey response samples, whether or not they supplied an anonymous ID.

Table . Racial Distribution

|  |  |  |
| --- | --- | --- |
|  | **Binary Search** | **Selection Sort** |
| Hispanic | 2 | 2 |
| Asian | 6 | 11 |
| Black | 3 | 3 |
| White | 14 | 22 |
| Mixed | 1 | 2 |
| North African | 0 | 1 |
| No Response | 9 | 12 |
| **Totals** | **35** | **53** |

Table . Gender Distribution

|  |  |  |
| --- | --- | --- |
|  | **Binary Search** | **Selection Sort** |
| Female | 10 | 19 |
| Male | 18 | 25 |
| No Response | 7 | 9 |
| **Totals** | **35** | **53** |

Table . Course and Section Distribution

|  |  |  |
| --- | --- | --- |
|  | **Binary Search** | **Selection Sort** |
| CS1-A.1 | 0 | 11 |
| CS1-A.2 | 17 | 16 |
| CS1-A.3 | 12 | 1 |
| CS1-B | 6 | 15 |
| CS2 | 0 | 10 |
| **Totals** | **35** | **53** |

## Technical Knowledge

Our primary technical goals for what the students would learn from the videos were being able to 1) identify the time complexity of the algorithm, 2) answer questions that require simulation of the algorithm, and 3) remember fundamental facts about the algorithm. With this in mind, we evaluated students’ baseline knowledge of the algorithm in the pre survey by asking if the students were familiar with big-O notation and if they were confident coding a search or a sorting algorithm. After watching the series of videos in order (intuition, analysis, song and code) we then asked one question per each of our goals in a multiple choice format where 1 of 5 possible selections was correct.

We now present a set of tables, each providing the student responses for the conceptual questions asked on the post-survey.

Below we can see that students were relatively unfamiliar with big-O notation with 9.7 % answering correctly for binary search and 15% for selection sort **[JD: I do not this we show this]**. However, after watching the videos students showed significant improvement in their knowledge of the algorithms: 85.7% of students were able to identify the runtime of binary search and 84.9% students were able to identify the runtime of selection sort. Additionally we can also see that the other questions were all answered correctly at least 80% of the time.

Table 4. Technical Understading Results

|  |  |
| --- | --- |
| **Question** | **% Correct** |
| Time complexity of Binary Search | 6.7 *pre* 85.7 *post* |
| Binary Search execution question | 82.9 |
| Binary Search conceptual question | 80.0 |
| Time complexity of Selection Sort | 14.9 *pre* 84.9 *post* |
| Selection Sort invariant question | 81.1 |
| Selection Sort conceptual question | 81.1 |

## Self-Gauged Competency

Since the majority of students who would watch these videos are beginners in computer science, a major goal was to measure students’ confidence with concepts such as complexity and big-O as well as to reinforce some underlying principles involved with algorithm design (*e.g.,* input parameters and preconditions, simple typing, control structures).  Each video series was presented approximately midway in each course (immediately before the algorithms were introduced in lecture), but served as an introduction to the ideas of algorithmic searching and sorting for many of the students. In the pre-survey where students were asked to grade their confidence in being able to code a sort the average response was 2.41 out of 5 and for coding a search the average response 1.77 out of 5. Likewise, when asked about big-O, students were also not very familiar with the concept with an average response of2.14 for selection sort and 1.54 for binary search**.** These values were understandably low, but we wanted to see how much students’ perceived confidence level improved as a result of following the video series.

Next, we wanted to see if there was a significant increase in perceived confidence. We conducted paired *t*-tests for each of four questions on perceived confidence, and the results are shown in Table 10.

Table . Student-Reported Confidence Increases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***df*** | ***p*** | **Mean increase** | **Mean percent increase** |
| Selection Sort coding | 49 | 6.21 x10-12 | 1.32 | 54.8% |
| Selection Sort complexity | 49 | 2.84x10-8 | 1.34 | 62.6% |
| Binary Search coding | 30 | 5.89x10-12 | 1.58 | 89.6% |
| Binary Search complexity | 30 | 2.41x10-11 | 1.68 | 109.7% |

For each question in Table 10 we find there is a significantly positive difference between pre and post responses for the entire sample. Moreover, since we were interested in how people from different demographics responded, we investigated if there was a difference in how much confidence gain there was between male and female students. To verify these results, we conducted a *t*-test for difference in means between two samples (male and female students) and did not consider anyone who was missing an ID or did not provide their gender on the survey. Below we see the mean difference in confidence per question per gender. For each question the mean confidence increase for women was higher, but none of the questions yielded a large enough spread between men and women for us to conclude there is a statistically significant difference in their confidence boosts. **[Table missing on gender]**

Table . Student-Reported Confidence Increases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Mean increase for men** | **Mean percent increase for men** | **Mean increase for women** | **Mean percent increase for women** |
| Selection Sort coding | 1.33 | 53.2% | 1.44 | 64.1% |
| Selection Sort complexity | 1.458 | 65.0% | 1.44 | 74.2% |
| Binary Search coding | 1.40 | 67.7% | 1.89 | 121.9% |
| Binary Search complexity | 1.53 | 78.9% | 2.11 | 211.0% |

## Critique of Videos

Students evaluated the videos using a Likert-type scale, where 1 means “strongly disagree,” 3 means “neutral” and 5 means “strongly agree.” Our main objectives in this section were to gain insight on the students’ experiences watching the videos, their attitudes after watching the videos, their responses to the music, and how they planned to utilize the videos. In order to ensure that students were thoroughly reading the questions before answering we added two questions with negative sentiments, in which “strongly agree” meant that they did not like a certain aspect of the videos. In addition to supplying the mean and standard deviation of each response, we also include p values for tests against the null hypothesis that the mean response was neutral.  Below are the results for both algorithms per question.

Table 12. Student Agreement Level with Statements. Likert Scale: 1 = Strongly Disagree, 3 = Neutral 5 = Strongly Agree

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Binary Search** | | | **Selection Sort** | | |
| **Video Series Statement** | *mean* | *st. dev.* | *P* | *mean* | *st. dev.* | *P* |
| Help remember | 2.771 | 1.00 | 0.186 | 2.603 | 0.95 | 0.004 |
| Not help understanding | 2.685 | 1.16 | 0.117 | 2.981 | 1.03 | 0.894 |
| Flowed naturally | 3.657 | 0.94 | <.001 | 4.094 | 0.79 | <.001 |
| Translated readily | 3.771 | 0.97 | <.001 | 3.641 | 1.08 | <.001 |
| Experience more fun | 3.314 | 1.16 | 0.117 | 3.320 | 1.19 | 0.0549 |
| Less intimidating | 3.657 | 1.21 | 0.003 | 3.735 | 0.96 | <.001 |
| Not a good use of my time | 2.714 | 1.12 | 0.16 | 2.377 | 1.02 | <.001 |
| Good study resource | 3.428 | 1.04 | 0.02 | 3.415 | 1.04 | 0.0056 |
| Want more videos series | 3.142 | 1.12 | 0.454 | 3.301 | 0.97 | 0.028 |

For *p* values, null:µ = 3

We can see that in both series students felt that the flow of the videos was logical and natural, the lyrics of the songs translated well to pseudo code, the videos overall made the algorithms less intimidating, and students felt that the videos would be a good study resource for exams. These questions had responses that were significantly more positive than neutral for both algorithms, but even if questions do render a close or even less than average response there can still be merit to the information they offer. For instance, when asked if they wanted more videos to help them learn other algorithms and concepts, selection sort had an average response rate of 3.30 and binary search had an average of 3.14, but for both algorithms over 41% of students indicated a score of 4 or 5, and under 21% of students indicated a score of 2 or 1. Although we did not expect them to memorize it after the first listen we still asked students if the song would help them memorize the algorithm and over 20% reported that they would for both algorithms.

## Free Responses

Below are all of the comments provided to us for both binary search and selection sort.

*Binary Search:*

* I suggest having a button somewhere that will play the song, and only the song, so I can play it over and over again without having to search through the video for it.
* This is great! The quality of the background piano music is fantastic.
* Great to learn concepts for prep before class.  Would be an awful source to study for exams.
* Generally for me, videos help more with review than the first time learning anyway, so I think it's a good review but I'm not sure about learning the first time.
* Good videos and good song, it presents the material in an entertaining way. But the videos were a little too long.
* The videos felt like they were moving a little slow to me. It's probably fine- it’s definitely better to err on the side of being slower anyway, and it’s easy enough to go to youtube and play them at double speed.
* If meant to watch all three (as in this was your first exposure) they became repetitive and not very time efficient.
* Catchier songs will probably be difficult to create but easier to memorize.
* It was great! A very inventive idea- I've never seen mnemonics applied to CS before. And it was really cool that you used your musical heritage to come up with the tune.
* I like how the videos took a pretty hard idea and made it simple.
* These videos would be good for a middle school or high school coding class. This is college. I don't like being spoon fed.

*Selection Sort:*

* Not sure if memorizing the song would be that helpful, but watching the videos—being able to pause, rewind, and fast forward at my own pace—certainly was.
* For the level of difficulty of the algorithm, the videos were excessively long and repetitive.
* I thought that they were interesting takes on Computer Science (that is, the inclusion of song lyrics and the interactive blackboard)!
* The song wasn't catchy enough to make it helpful. Try a catchier tune.
* The videos had good information, but went a little too slowly. I would prefer a combination of text and pictures to videos.
* This is great! The quality of the background piano music is fantastic.
* These videos may have assumed a bit too little background knowledge. They were helpful though, thanks!
* …I like my information to be straight to the point and clear when learning, so I might not be your intended audience. They are not horrible by any means, just not great and not suited to my personal learning style.
* The song didn't help me learn the algorithm, I learned what it was before I got to the song. However, it will likely help me out in memorizing it.

For the most part we can see that some people liked the music, others did not. However for those who did like the song, comment X in selection sort was very insightful and would allow users to interact with the material exactly the way we intend. The song, like any piece of music is to be repeatedly rehearsed until it can be recited from memory. By practicing the song though students are employing a varied study strategy since they are using music and voice which activates different parts of their brain not typically stimulated during studying. Having a varied approach like this is known to reinforce concepts powerfully and create a deeper understanding of the material since students are manipulating information in different ways and making connections through a wider array of locations in the brain [Roediger]. Other useful information that we gathered from these comments were suggestions to try writing pseudocode to pieces that students already know, which could be an interesting point of comparison to the existing songs.

One common theme in the comments were that the videos were too long, and could be conveyed more concisely. However, as comment X notes in binary search, it is better to err on the side of too long in order that students who do need to see a few more explanations before something clicks, can benefit from the videos, too. This is a valid concern though and in order to address it we need to convey to students that it is permissible to watch the videos at 1.5 or double speed and/or skip over some parts once they have accomplished the video objectives (stated at the beginning of the video). Moreover studying students’ responses to shorter videos would be a worthwhile research endeavor to investigate if the concepts resonate as well as they did with relatively longer videos. Lastly, although it may have appeared as disapproving feedback, comment number X does address a possible area of future research in assessing how well high school and even middle school students respond to the video series as an introduction to algorithms.

In addition to some of the ideas we have gathered from student free response input, we have also considered ways to augment the videos. Musically, topics of study could be the effect of adding different instruments to the music, or different singers, composing with different styles of music (although the styles of the two songs in this study were quite different), and writing various lengths of song (they are kept purposely short as it is now) . In addition to changing the content itself we have also considered different ways of changing the methodology employed in this study

# CONCLUDING DISCUSSION

Educators at all levels are under intense, often increasing, pressure to cover a complete curriculum inside of a limited number of lectures. This tension makes it difficult to develop interactive and meaningful learning opportunities as opposed to lecture-only delivery of material. There is just too much information to cover in too little time.

Another problem that we in computer scienceare facing today is diversity throughout all levels of study. We want to provide equity of access to the material. Music, we believe, provides an almost universal means to reach students and engage them. This project uses music to promote learning about basic algorithms.

We propose a video series used to teach algorithms that incorporates the emotional and mnemonic power of music, and employs audio-visual demonstrations to teach concepts and guide students through such topics as sorting, searching, and complexity analysis.

The study covered in this paper, which was held at two colleges across three different courses with five different instructors, has provided evidence to that students gained a strong conceptual grasp of the material which they had little to no understanding before by watching these video series. Moreover, when testing students’ attitudes of the material, results show that their baseline level of confidence increased significantly. This increase in confidence was seen in both males and females, but was actually higher for female students.

The core message is that there is nothing to lose from using these video series and so much to gain. Having 80% of students walking into a classroom understanding concepts such as big-O, searching and sorting to some degree prior to the meeting can open up class time for teachers. This will enable them to include stimulating and productive learning activities such as in-class problem solving time and peer instruction. These video series are freely and readily available, allow students to work at their own pace, are accessible to anyone with Internet access, and even incorporate the power of music to illustrate code. Thus, we encourage other educators to explore these videos and to offer the series to their students, especially novice students, as a fun, effective way to develop their appreciation and understanding of computer science.

# ACKNOWLEDGMENTS

Our thanks to the professors who piloted the videos series in their respective courses in the Spring of 2016.

# REFERENCES

1. Ames, B.J. 2012. What is a flame? The Flame Challenge: Explaining Science to an 11 Year Old. *Alan Alda Center for Communicating Science, Stony Brook University*. Accessed 20 August 2016, <http://www.centerforcommunicatingscience.org/what-is-a-flame/>.
2. Bransford, J.D., Brown, A.L. and Cocking, R.R., 1999. *How people learn: Brain, mind, experience, and school*. National Academy Press.
3. Chavey, D. 1996. Songs and the analysis of algorithms. In *Proceedings of the Twenty-seventh SIGCSE Technical Symposium on Computer Science Education*, Karl J. Klee (Ed.). ACM, New York, NY, USA, 4 – 8.
4. Claussen, D. W., and M. H. Thaut. 1997. “Music as a mnemonic device for children with learning disabilities.” *Canadian Journal of Music Therapy* 5.1: 55 – 66.
5. **Dougherty, J. P. 2008. “Using lyrics and music to reinforce concepts.” *Journal of Computing Sciences in Colleges*, 23, 3 (Jan. 2008), 106 – 113.**
6. Knuth, D. 1984. The complexity of songs. *Commun. ACM* 27(4), 344 – 346.
7. Maher, M.L., Latulipe, C., Lipford, H., and Rorrer, A. 2015. “Flipped Classroom Strategies for CS Education.” *In Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. ACM, New York, NY, USA, 218 – 223.
8. **Schreiber, B., and Dougherty, J. P. 2016. “Embedding Algorithm Pseudocode in Lyrics to Facilitate Recall and Promote Learning.” Accepted, *Consortium for Computing Sciences in Colleges Eastern Conference* (Frostburg State University, Frostburg, MD, USA, October 28 – 29).**
9. Yeoh, M.P. 2013. “Musical mnemonics to facilitate learning of matriculation biology: The Calvin Cycle.” *Molecules* 3: 3C.
10. Zull, J.E., 2004. The art of changing the brain. *Educational Leadership*, *62*(1), pp.68-72.

Columns on Last Page Should Be Made As Close As Possible to Equal Length

* **HomerM. L. West**, The singing of Homer and the modes of early Greek music, Journal of Hellenic Studies 101 (1981), 113-129. [General considerations; the tuning of the phorminx (but see: Ancient Greek Music, Oxford 1992, 328). The given melodies do not conform to our theory.] http://www.oeaw.ac.at/kal/sh/

] Thaut, M.H., Peterson, D.A., and McIntosh, G.C. “Temporal entrainment of cognitive functions.” *Annals of the New York Academy of Sciences* 1060.1: 243-254, 2005.