Research on the Education of Scientific Paper Writing for Junior College Students: A Case Study of C++ Language Program Development

Xiaofu Du

Huzhou Vocational and Technical College, Huzhou Key Laboratory of IoT Intelligent System Integration Technology(2022-21), Huzhou, 313099, China

4612889@qq.com

Abstract. To improve the writing ability of junior college students, this study is carried out. First of all, the guiding ideology of project drive is put forward to exercise students' writing ability in the actual project learning. Secondly, more detailed guidance and essay models should be provided to give students clearer instructions. Finally, the requirement of innovation should be reduced and the teaching of junior college paper writing should be based on confirmatory experiments. In the course of C++ Language Program Development, several appropriate topics were summarized to guide students through a sample essay. The practice results show that junior college students have higher acceptance of the above teaching methods and get better learning effect.

Keywords: Junior college students, Technical paper writing, Sorting algorithm, Time efficiency.

1 Introduction

Scientific and technical paper writing is highly valued in graduate education [1] and undergraduate education [2], especially in many science and technology majors, but in most junior colleges, scientific and technical paper writing is not valued.

However, with the development of The Times, junior college students are also faced with the need to write scientific and technological reports and even scientific research papers in their study and future work. For example, in recent years, in many discipline competitions, junior college students are required to complete the writing of scientific research papers. For another example, many courses now increasingly use practical assessment methods, among which writing papers for course content has become a very popular assessment method. As a result, it has become a realistic and urgent need to carry out science and technology paper writing education for junior college students.

There are few researches on the writing of scientific and technological papers for junior college students. Jin Li[3] discusses the necessity and characteristics of science and technology paper writing education in junior colleges from the aspects of thesis topic selection, thesis writing common sense and training system. Fen Lou[4] discusses

F. Zeng et al. (eds.), *Proceedings of the 2024 7th International Conference on Humanities Education and Social Sciences (ICHESS 2024)*, Advances in Social Science, Education and Humanities Research 887,
https://doi.org/10.2991/978-2-38476-323-8_19

the characteristics of junior college students' thesis writing from the aspects of thesis topic selection, data collection, thesis writing, thesis defense, etc.. Yanxia Zhao[5] discusses the topic selection, literature review, writing level and so on, and proposes that it should start from the "small" research, start from the "Pre-school education research method" course to effectively train students in writing papers, and improve their graduation thesis work level.

According to the author's many years of work experience in colleges and universities, the thesis writing of junior college students has the following characteristics compared with undergraduate and graduate students.

1) Writing for different purposes. Junior college students during the study of the main purpose of writing papers is to complete the course work or assessment, participate in the discipline competition, complete the graduation thesis. However, the purpose of undergraduate and postgraduate paper writing is more extensive, and it is more to sort out the results of academic activities.

2) The different purpose of writing leads to the low requirement for innovation in the writing process of junior college students. The topic selection of junior college students' thesis writing is more verification experiment, which requires higher repeatability of the paper content.

3) Junior college students are not familiar with the writing norms of scientific and technological papers, which leads to junior college students having no way to start when they write for the first time.

In view of the above characteristics, the education of scientific and technological paper writing for junior college students should also be targeted.

1) Guide students to choose a reasonable topic. Starting with confirmatory experiments, guide students to familiarize themselves with the basic methods of essay writing. In the later stage, gradually guide the competent students to put forward their own innovative topics.

2) Teaching focuses on the general composition, format and writing method of scientific and technological papers, so that students can learn to use papers to describe ideas and organize results as soon as possible.

I teach software technology major, the following "C++ language program development" course as an example, to discuss the characteristics and methods of junior college students science and technology paper writing education. In the course of "C++ Language Program Development", students need to deeply understand many algorithms, students can choose the sorting algorithm as one of the thesis topics. For example, in the following, the time efficiency of common sorting algorithms will be analyzed. The following papers are mainly divided into six parts: Introduction, Related Work, Method, Experiment, and Conclusion, which clearly introduce the common paradigms of scientific and technological paper writing to students.

2 Summary

According to the literature, the time efficiency of common sorting algorithms is basically divided into three categories: the time efficiency of selection sorting, bubble sorting and insertion sorting is O(n2); the time efficiency of quick sorting, heap sort and merge sorting is $O(n \log_2 n)$; and the time efficiency of bucket sorting is O(n).

In order to verify the correctness of the above statement and further understand the time efficiency of common sorting algorithms, the author completed this paper. Firstly, the author uses C++ language to implement the above three sorting algorithms, and then puts forward some innovations of his own.

1) Write test code to automatically test each sort algorithm. Each algorithm was tested 50 times, depending on the size of the problem, in order to get more adequate result data for later analysis.

2) Using chrono class library [6], write code to realize microsecond timing, and time the actual operation effect of three common algorithms, which is convenient for later analysis.

3) The freopen function is used to redirect the standard output stream object stdout to achieve the purpose of outputing the program output results to csv files. In this way, the program can be used to directly obtain data files that can be analyzed in accordance with the format requirements.

4) Finally, use EXCEL to open the obtained CSV file, and use EXCEL formula and image functions to visually analyze the data file.

Using the above innovation points, we get intuitive experimental results, and the final experimental results also verify that the time efficiency of selection sort, quick sort and bucket sort is indeed, respectively.

In the "Introduction" chapter, complete the introduction of the topic background.

3 Related Works

In the "related works" chapter, the previous work related to the topic is generally carried out, and a detailed literature review is generally done in this chapter. For junior college students, it is necessary to appropriately reduce the difficulty of literature and focus on discussing the basic knowledge related to the paper.

This paper selects three sorting algorithms, their basic ideas are as follows.

3.1 Selection Sort

The complete selection sorting algorithm flow chart is shown in Figure 1.The selection sort consists of two loops, the outer loop finding a maximum or minimum value at a time, and the inner loop comparing all elements.

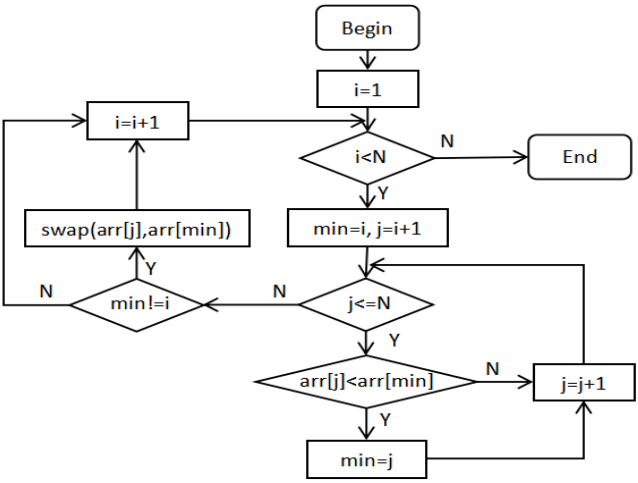

Fig. 1. Algorithm chart of selection sorting.

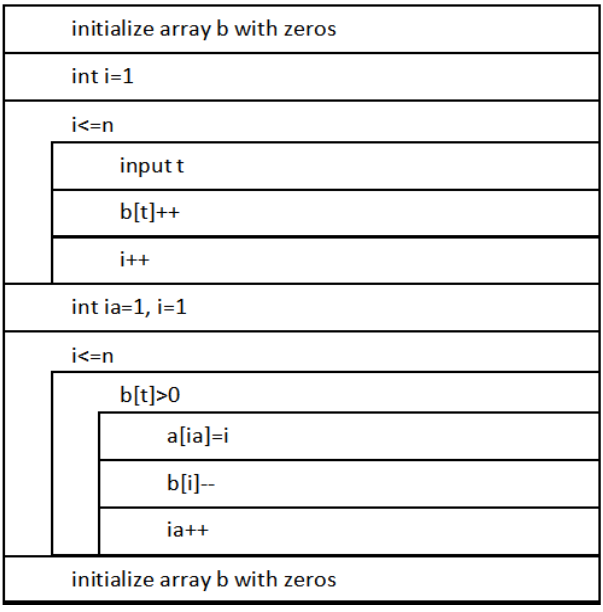

Fig. 2. Algorithm chart of bucket sort.

3.2 Quick Sort

In essence, quick sort is a recursive algorithm, which adopts the idea of divide and conquer, that is, select a base element mid in an unordered sequence, and use the mid to divide the sequence to be sorted into two parts, the first part of the elements are less than or equal to the base element mid , and the latter part is greater than or equal to the

base element mid , and then repeat the above operations for the two parts respectively by recursive method. Until the disordered sequence is arranged into an ordered sequence.

3.3 Bucket Sort

Bucket sorting is one of the fastest sorting algorithms, the basic idea is to sort and input data at the same time. Bucket sorting has a premise, that is, all the data to be sorted must belong to a not too large area, which can be called the range. The NS box diagram of bucket sort is shown in Figure 2.

4 Methods

In general paper writing, this chapter needs to describe the author's innovative approach. However, for college students, innovation requirements can be appropriately reduced. It is only necessary to clearly describe the author's thinking in conducting the experiment. The following is a sample passage。

After writing the code of the three sorting algorithms, in order to better analyze the time efficiency of the three algorithms, the following method improvements were made.

4.1 Test Code

First, I wrote test code that can automatically test each sort algorithm. Each algorithm was tested 50 times, depending on the size of the problem, in order to get more adequate result data for later analysis.

During the testing process, I need to constantly adjust the value of the data size N in the test. In practice, my test size for selection sort starts at N=10000 and increases by 2000 each time until N=100000; Test sizes for Quicksort and bucket sort start at N=10 million and increase by 2 million each time until N=10 million. The reason I chose the two test schemes that differ by a factor of 1000 is that I found the selection sort time consumption to be much greater than Quicksort and bucket sort. If I use the same scale for all three sorts, then either the selection sort is too slow or the other two sort is too fast to do good data analysis. Of course, in order to compare the three algorithms more intuitively, I also set up a set of test scales to let the three algorithms compete on the same stage.

4.2 Timing Code

After querying the documents, chrono class library [6] is used to write code to realize microsecond timing, and the actual running effect of three common algorithms is timed for later analysis.

It should be noted that because the chrono class library is a relatively new class library, and the C++ development tool I use is DEV C++, I need to set the compiler version to C++11, and there will be compilation errors.

4.3 Data File Generation

In order to better analysis, I use freopen function to redirect the standard output stream object stdout, and realize the purpose of output the program output result to csv file. In this way, the program can be used to directly obtain data files that can be analyzed in accordance with the format requirements.

The CSV file was chosen because of its simple format: each line ends with a return character, and the different columns in each row are separated by a comma ", ".

4.4 Excel Data Analysis

Finally, the obtained CSV file is opened with Excel, and the data file is visually analyzed by using EXCEL formula and image functions. Excel's data formulas and charts are powerful enough to be used as a data analysis tool.

It should be noted that for comparison, a standard time efficiency function curve needs to be plotted. At this time, it is necessary to adjust the coefficient of the function properly to make the image of the standard function better coincide with the fitting curve of the actual running time result.

5 Experiments

In this chapter, the experimental results are generally analyzed. The following is a sample passage.

According to the previous steps, we completed the development of all the code programs, and after the debugging was correct, I conducted several experiments on the three sorting algorithms. In each experiment, test data of different scales were used, and the experimental data as follows were finally obtained. I made the following analysis on them.

5.1 Comparison of Time Efficiency of Three Algorithms

To visualize the time efficiency of the three algorithms, we use the same problem size to test the performance of the three algorithms. Table 1 lists the final data.

Table 1. Time consumption of three algorithms for sorting data on different problem sizes (unit: microsecond)

Problem Sizes n	Selection Sort	Quick Sort	Bucket Sort
5000	10965	1019	0
10000	40047	0	997

15000	69315	1052	1016
20000	207994	0	3331
25000	298342	0	1993
30000	292961	0	2243
35000	456683	1478	3205
40000	528355	997	3987
45000	649213	0	3932
50000	790603	994	4984

The analysis of the above Table is shown in Figure 3.

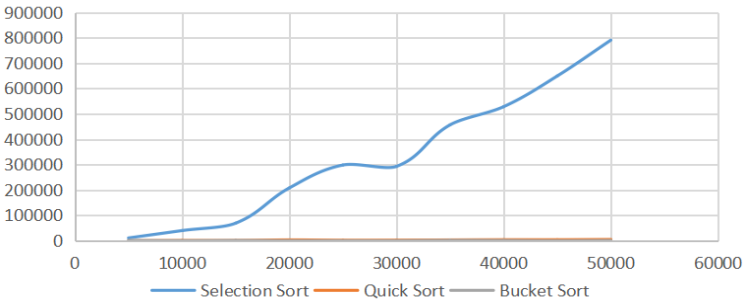

Fig. 3. Comparison of time efficiency of three algorithms.

It can be seen that as the problem size increases, the time consumed by selection sort also increases sharply, while quick sort and bucket sort increase relatively little. From Table 1, we can even see that many of the data for bucket sorting is 0 because bucket sorting takes too little time for the current size of the problem to even be measured by the code.

5.2 Time Efficiency Analysis of Three Sort Algorithm

We tested the performance of the three sorting algorithm using 50 different problem sizes. The final result data analysis is shown in Figure 4, Figure 5 and Figure 6.

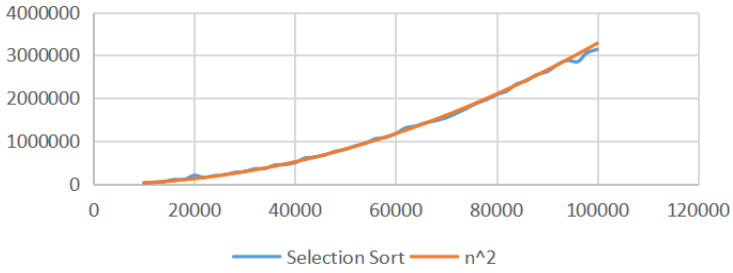

Fig. 4. Time efficiency analysis of selection sort algorithm

In Figure 4, we fit the data using the connected scatter plot, and the blue curve is the fitting result of the real data. Then, for comparison, we draw a theoretical curve for $O(n^2)$, which is represented by the red curve. From the result, the red and blue curves fit together very well. It shows that our experimental results verify that the time efficiency of selection sorting is indeed $O(n^2)$.

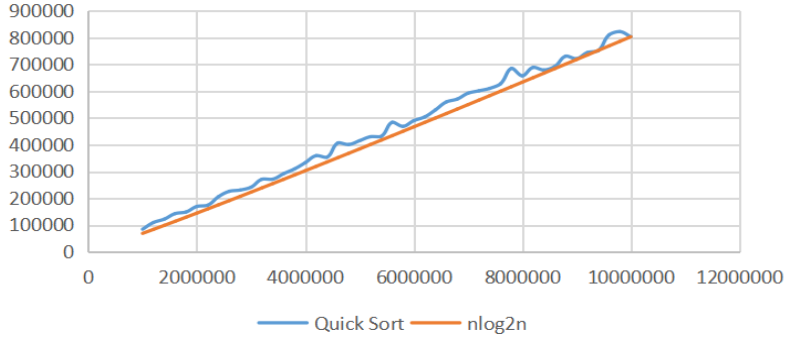

Fig. 5. Time efficiency analysis of quick sort algorithm

In Figure 5, we fit the data using a connected scatter plot, and the blue curve is the fitting result of the real data. Then, for comparison, we draw a theoretical curve for $O(n\log_2 n)$, which is represented by the red curve. From the result, the red and blue curves fit together very well. It shows that our experimental results verify that the time efficiency of quick sort is indeed $O(n\log_2 n)$.

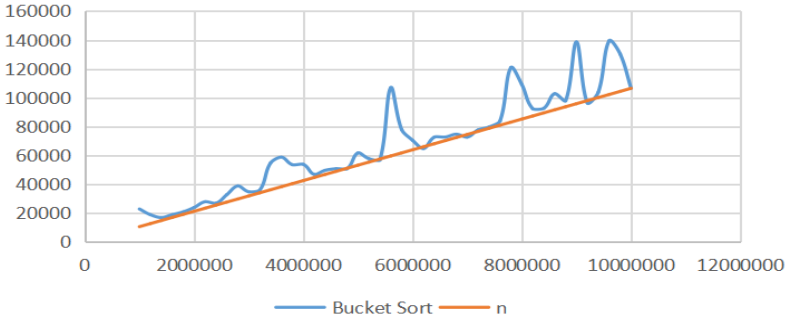

Fig. 6. Time efficiency analysis of bucket sort algorithm

In Figure 6, we fit the data using a connected scatter plot, and the blue curve is the fitting result of the real data. Then, for comparison, we draw a theoretical curve for $O(n)$, which is represented by the red curve. From the result, the red and blue curves fit together very well. It shows that our experimental results verify that the time efficiency of bucket sort is indeed $O(n)$.

In addition, we can see from Figure 6 that the jitter of the blue curve is relatively serious, which indicates that the performance of bucket sorting in actual operation is

not stable, and is affected by factors such as problem scale, whether the original data is partially ordered, and the running state of the computer.

6 Conclusions

For technical paper writing, the "introduction" and "conclusion" sections are the focus, and most readers will get a quick overview in these two sections. The following is a sample passage.

In this paper, the author repeats the three sorting algorithms, and uses the test data of different problem sizes to test, and finally completes the data analysis. The final experimental results also verify that the time efficiency of selection sort, quick sort and bucket sort is $O(n^2)$, $0(n\log_2 n)$and $O(n)$ respectively.

According to the experimental results, bucket sorting has the highest efficiency, but it has the disadvantage of unstable performance. In addition, according to the principle of bucket sorting, bucket sorting is only suitable for the original data distribution in a fixed area. Bucket sorting cannot be used if the original data is not guaranteed to be distributed within a certain region.

From the perspective of universality, the best sorting algorithm is quick sorting, especially in the case of a large amount of data, the speed is much faster than the selection sorting algorithm. Of course, selection sort and bubble sort also have their advantages. If the original data is basically ordered and only a small amount of data needs to be restructured, the performance of selection sort and bubble sort will be better than quick sort.

In this paper, we only discuss the basic sorting algorithms, and we do not discuss the special sorting algorithms such as multiple merge sorting. In the subsequent learning process, the author will further study various sorting algorithms.

Acknowledgment

In the "Acknowledgments" section, thank all the people, organizations, or funds that contributed to the writing of the article.This work was supported by the Huzhou Vocational and Technical College Special Topic For High-Level Talents (2023TS11).

References

1. Jiaxing Wang, Bin Cao, Xusheng Yang and Jing Fan. (2024) Exploration of ideological an d political teaching reform in the course of computer science paper writing for graduate stu dents. Computer Education, 08:50-54. DOI:10.16512/j.cnki.jsjjy.2024.08.043.
2. Li Jialin, Ying Chao, Cao Luodan, et al. (2024) Exploration on Ideological and political tea ching of undergraduate "Scientific Research Paper Writing" course based on OBE concept. Journal of ningbo institute of education, 26 (01) : 109-114. DOI: 10.13970 / j.carol carroll nki nbjyxyxb. 2024.01.025.
3. Jin Li. (1996) How to guide junior college students to write scientific papers. Journal of Da lian Institute of Education, 01:49-50. https://kns.cnki.net/kcms2/article/abstract?v=3uoqIh

G8C44YLTlOAiTRKjkpgKvIT9NkZNmQNo4kSVodEUgeW-KQi5mXZMmX9pxkxaKEV6Y-cvx3Dpm9HENa5en5M4q2zZ2S&uniplatform=NZKPT.

4. Fen Lou. (2005) A brief discussion on the characteristics and guiding skills of graduation thesis writing for amateur junior college students. Science & Technology Information, 26: 189. DOI:10.16661/j.cnki.1672-3791.2005.26.155.
5. Yanxia Zhao. (2021) The problems and solutions in the graduation thesis work of Preschool Education major -- A case study of Ancient Preschool Teachers College in Inner Mongolia. Science Tribune, 14 : 62-64. DOI: 10.16400 / j.carol carroll nki KJDK. 2021.14.020.
6. Foam o0,2024. C++ std::chrono library usage Guide. https://blog.csdn.net/qq_21438461/article/details/131198438?ops_request_misc=%257B%2522request%255Fid%2522%253A%252208D91355-1E11-4666-AECB-113EDD6E95CE%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=08D91355-1E11-4666-AECB-113EDD6E95CE&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-1-131198438-null-null.142^v100^pc_search_result_base2&utm_term=chrono%20%E8%AE%A1%E6%97%B6&spm=1018.2226.3001.4187.