

ISSN 2181-8622

Manufacturing technology problems



Scientific and Technical Journal Namangan Institute of Engineering and Technology

INDEX  COPERNICUS
INTERNATIONAL

**Volume 9
Issue 1
2024**



ANALYSIS OF USE OF SORTING ALGORITHMS IN DATA PROCESSING

TUYCHIBAEV HAMIDULLO

Master student of Namangan Engineering-Construction Institute, Namangan, Uzbekistan

Tel: (0890) 067-2273, E-mail: tuychibaevhamidullo1999@gmail.com

Abstract. This article aims to identify the most efficient sorting algorithm for solving array problems in the programming language C++. By analyzing existing sorting algorithms and conducting performance tests, we can determine the algorithm that best meets the requirements based on the analysis of test results. Analyzing the processor load data obtained during the execution of the test tasks performed using the bubble-sort method and the selection method; we can consider the selection-sorting algorithm as the most optimal algorithm for specific test tasks. Tasks performed on it, the processor used the least resources. It is not possible to speak for sure about the efficiency of this method, since the number of used test tasks was relatively small, but the obtained experimental results allow us to conclude that the selection-sorting algorithm is more efficient in terms of the use of processor resources. After careful analysis and performance testing, we have successfully identified the most efficient sorting algorithm for solving array tasks in C++. By choosing this algorithm, developers can optimize their code and improve the overall performance of their applications. It is important to consider specific requirements and constraints when choosing a sorting algorithm, but this study provides a solid basis for making an informed decision.

Keywords: programming, programming languages, data structures, sorting algorithms, data, testing tasks, the results of test tasks, optimization, processor, the efficiency of the method.

Introduction. Correct and effective use of sorting algorithms in data processing is very important. When working with a large amount of data, the correctly chosen method ensures fast and efficient operation of the software product. In addition to choosing a programming language for the qualitative formation of algorithmic thinking and obtaining competencies in the specialty, the students of the specialized specialty are taught the basics of algorithmization and various algorithms for solving problems. The most common and important are data sorting algorithms, which are often used in solving computational problems, including working with data arrays. Today, there are several data sorting algorithms. The most common and most used ones include Bubble Sort Algorithm, Quick Sort Algorithm, Shell Sort Method, and Selection Sort Algorithm.

Research methodology. The purpose of this work is to analyze two sorting algorithms ("bubble sort" algorithm and selection sort algorithm) and determine the most optimal one for solving the tasks of computing data arrays in the C++ programming language. In order to evaluate the effectiveness of the selected algorithms ("bubble" and selection), it was decided to use them in solving a number of test tasks. In addition, based on the obtained results of the processor load in the calculation of these tasks, a conclusion was drawn about the effectiveness of a certain sorting algorithm.

Analysis and results. The following tasks were selected as test tasks:

Task 1. Given a one-dimensional array. Sort the array in ascending order.

Task 2. Sort a two-dimensional array in ascending order.

The basic code structure for Task 1 using the bubble sort algorithm is as follows:

```
int main(int argc, char* argv[])  
{  
    srand(time(NULL));
```

```
cout << " Enter the size of the array: ";
int D;
cin >> D; // Entering the size of a function
int *massiv = new int [D];
for (int i = 0; i < D; i++) // Filling an array with random data
{
    massiv[i] = rand() % 100 + 1;
    cout << setw(2) << massiv[i] << " ";
}
cout << "\n\n";
cout << " Sorted array: "<<endl;
for (int i=D-1;i>0;i--) // Bubble-sorting
{
    for (int j=0;j<i;j++)
    {
        if(massiv[j]>massiv[j+1])
        {
            int t=massiv[j];
            massiv[j]=massiv[j+1];
            massiv[j+1]=t;
        }
    }
}
for (int i = 0; i < D; i++)// Array output
{
    cout << setw(2) << massiv[i] << " ";
}
cout << "\n\n";
system("pause");
return 0;
}
```

The following indicators of processor load were obtained when executing the program code of task 1, written in the C++ programming language, using the above-mentioned bubble-sorting algorithm (Fig. 1):



Figure 1. CPU load when performing task No. 1 using the sorting algorithm via the bubble-sorting algorithm.

The text of the program for the implementation of the first task using the sorting algorithm with the selection method:

```
void choicesSort(int*, int);
int main(int argc, char* argv[])
{
    srand(time(NULL));
    cout << " Enter the size of the array: ";
    int D;
    cin >> D; // Entering the size of a function
    int *massiv = new int [D];
    for (int i = 0; i < D; i++) // Filling an array with random data
    {
        massiv[i] = rand() % 100 + 1;
        cout << setw(2) << massiv[i] << " ";
    }
    cout << "\n\n";
    choicesSort(massiv, D); // Calling the sort function
    for (int i = 0; i < D; i++) // Display an array on the screen
    {
        cout << setw(2) << massiv[i] << " ";
    }
    cout << "\n";
    system("pause");
    return 0;
}
void choicesSort(int* arrayPtr, int length_array) // Implementation of the selection
function
{
    for (int repeat_counter = 0; repeat_counter < length_array; repeat_counter++)
```

```

{
    int temp = arrayPtr[0];
    for (int element_counter =
        repeat_counter + 1; element_counter <
        length_array; element_counter++)
    {
        if (arrayPtr[repeat_counter] > arrayPtr[element_counter])
        {
            temp = arrayPtr[repeat_counter];
            arrayPtr[repeat_counter] =
            arrayPtr[element_counter]; arrayPtr[element_counter]
            = temp;
        }
    }
}
}
}

```

The following processor load indicators were obtained when executing the program code of the first task written using the sorting algorithm with the selection method (Fig. 2).

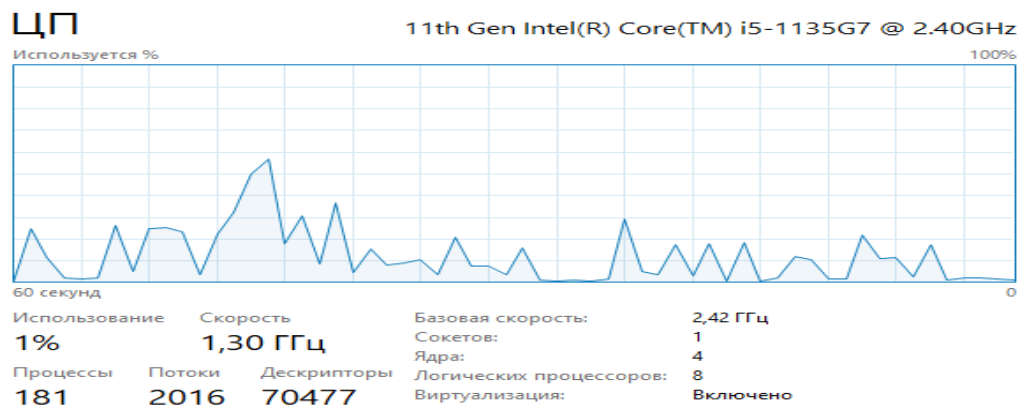


Figure 2. CPU load when performing task No. 1 using the sorting algorithm via the selection method.

Based on the obtained data, it is possible to construct a graph of the processor load when calculating task No. 1 using different sorting algorithms (Fig. 3). Based on the obtained data, we can conclude that the selection method algorithm is more suitable for solving task 1.

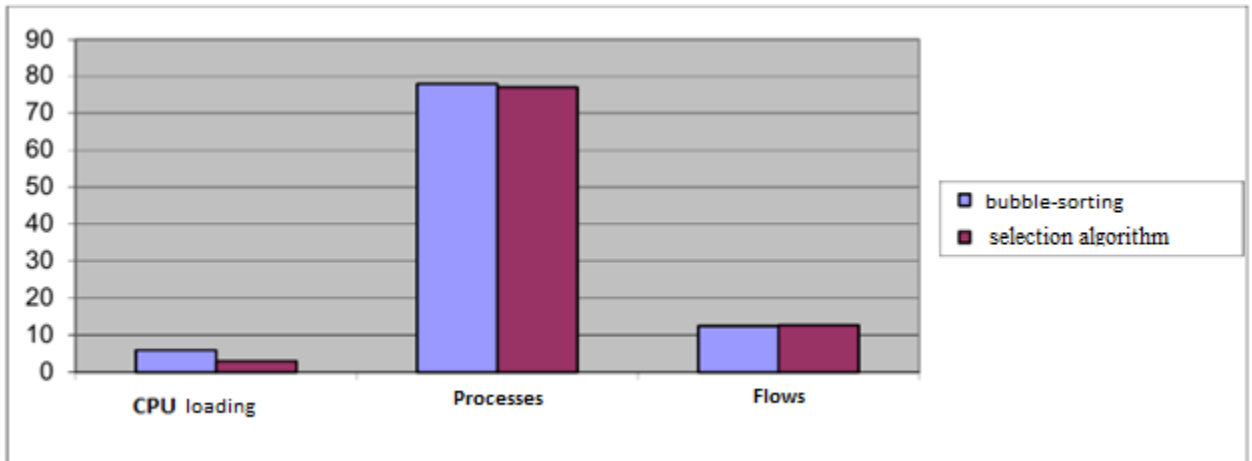


Figure 3. Comparison graph of CPU load when computing task No. 1 using different sorting algorithms.

Two sorting algorithms were also used in the implementation of test task No. 2: sorting by "bubble" and selection methods. Below is the basic code structure for Task 2 using the bubble-sorting algorithm:

```
int main()
{
    const int D = 5;
    unsigned short k;
    int i, j, i_min, sum;
    int massiv[D][D];
    srand(time(NULL));
    for(i = 0; i < D; ++i) // Filling a two-dimensional array with random data
    {
        for(j = 0; j < D; ++j)
        {
            massiv[i][j] = rand()%50 + 1;
        }
    }
    cout<<" Initial matrix: "<<endl;
    for(i = 0; i < D; ++i) // Display the initial matrix on the screen
    {
        for(j = 0; j < D; ++j)
        {
            cout << massiv[i][j] << '\t';
        }
        cout << endl;
    }
    cout << " Sorted array: "<<endl;
}
```

```
for(int c=0;c<=(i*j);c++)// A bubble to sort the matrix in ascending order
{
    for(int i2=0;i2<i;i2++)
    {
        for(int j2=0;j2<j-1;j2++)
        {
            if(massiv[i2][j2]>massiv[i2][j2+1])
            {
                int temp=massiv[i2][j2];
                massiv[i2][j2]=massiv[i2][j2+1];
                massiv[i2][j2+1]=temp;
            }
        }
    }
    for(int x=0;x<j;x++)
    {
        for(int z=0;z<i-1;z++)
        {
            if(massiv[z][x]>massiv[z][x+1])
            {
                int temp=massiv[z][x];
                massiv[z][x]=massiv[z][x+1];
                massiv[z][x+1]=temp;
            }
        }
    }
}
for(int i2=0;i2<i;i2++) // Display the matrix on the screen
{
    for(int j2=0;j2<j;j2++)
    {
        cout<<massiv[i2][j2]<<"\t";
    }
    cout<<"\n";
}
return 0;
}
```

The following indicators of processor load were obtained when executing the program code of task 2 written in the C++ programming language using the bubble-sorting algorithm (Fig. 4).



Figure 4. CPU load when performing task No. 2 using the bubble-sorting algorithm.

The basic code structure for the implementation of task No. 2 using the choice sorting algorithm:

```
int main()
{
    const int D = 5;
    unsigned short k;
    int i, j, i_min, sum;
    int massiv[D][D];
    srand(time(NULL));
    for(i=0;i<D;++i) // Filling a two-dimensional array with random data
    {
        for(j = 0; j < D; ++j)
        {
            massiv[i][j] = rand()%50 + 1;
        }
    }
    cout<<" Initial matrix: "<<endl;
    for(i = 0; i < D; ++i) // Display the initial matrix
    {
        for(j = 0; j < D; ++j)
        {
            cout << massiv[i][j] << '\t';
        }
        cout << endl;
    }
    int t1;
    unsigned short minind,
    mm = D - 1;
    for(i=0;i<D;i++) // Sorting the matrix in ascending order
    {
```



```

for (j = 0; j < mm; j++)
{
    minind = j;
    for (k = j+1; k < D; k++)
    if (massiv[i][minind] > massiv[i][k])
        minind = k;
    t1 = massiv[i][j];
    massiv[i][j] = massiv[i][minind];
    massiv[i][minind] = t1;
}
}
cout << " Sorted array: " << endl;
for(i = 0; i < D; ++i) // Display a sorted matrix
{

    for(j = 0; j < D; ++j)
    {
        cout << massiv[i][j] << '\t';
    }
    cout << endl;
}
return 0;
}

```

The following indicators of the processor load were obtained when executing the program code of the second task written using the sorting algorithm by the selection method (Fig. 5).

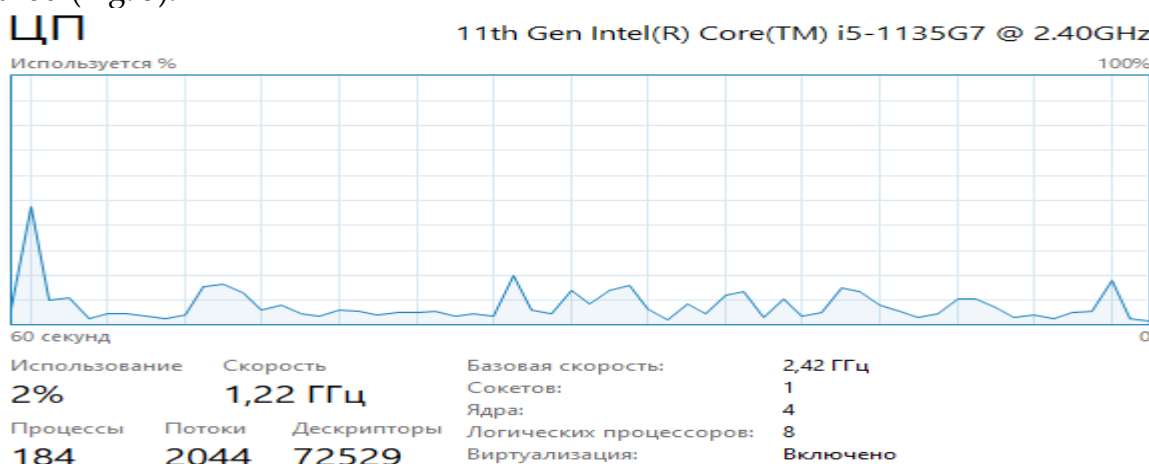


Figure 5. CPU load when performing task No. 2 using the selection algorithm.

Based on the obtained data, we can construct a graph of the processor load when calculating test task No. 2 using different sorting algorithms (Fig. 6).

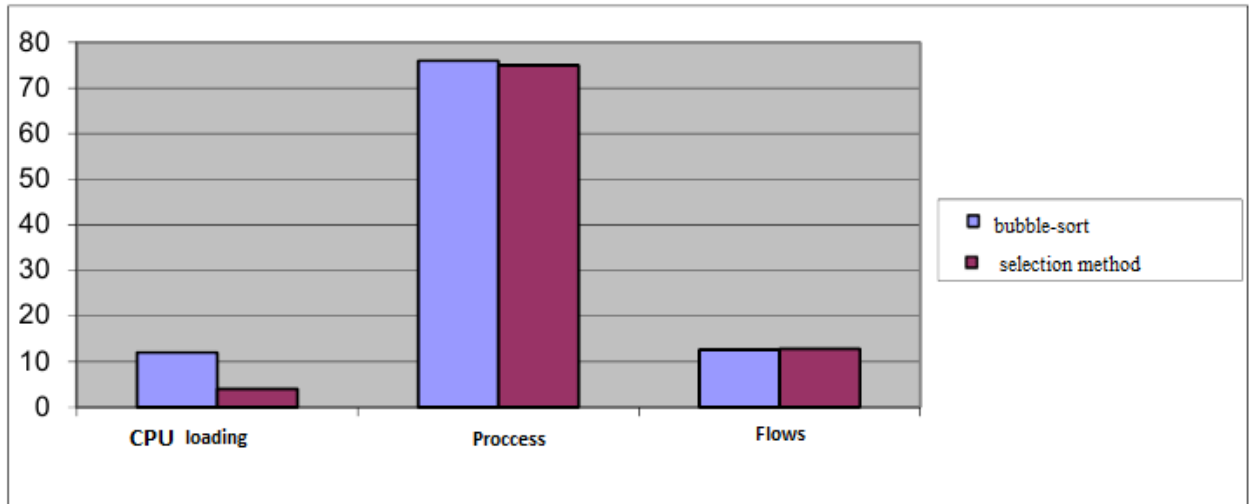


Figure 6. Comparison graph of CPU load when computing task No. 2 using different sorting algorithms.

Based on the obtained data, we can conclude that the selection method is the most effective for solving this task.

REFERENCES

1. Shildt, G. Self-instruction manual C++ / trans. from English – St. Petersburg: BHV- St. Petersburg, 1998. - 620 p.
2. Jamsa, K. Learning to program in C++ / translated. from English – M.: Mir, 1997. - 320 p.
3. Stephen Prata C++ programming language. Lectures and exercises. 6th ed. - M.: Williams, 2012.
4. Savitch, W. C++ in examples / trans. from English – M.: EKOM, 1997. - 736 p.
5. Deitel, H. Deitel, P. How to program in C++ / trans. from English – M.: BINOM, 1998. - 1024 p.
6. Rao, S. Master C++ on your own in 21 days. - 7th ed. – M.: Williams, 2013.
7. Stroustrup, Bjorn. C++ programming language. Special edition / trans. from English – M.: Binom, 2011. - 1136 p.
8. Stroustrup, Bjorn. Programming. Principles and practice of using C++ / trans. from English – M.: Williams, 2011. - 1248 p.
9. Laforet, R. Object-oriented programming in C++ / trans. from English – St. Petersburg: Peter, 2012. - 900 p.

C O N T E N T S

PRIMARY PROCESSING OF COTTON, TEXTILE AND LIGHT INDUSTRY

Nabidjanova N., Azimova S.	
Study of physical-mechanical properties of fabrics used for men's outer knit assortment	3
Nabidjanova N., Azimova S.	
Development of model lines of men's top knitting assortment	7
Noorullah S., Juraeva G., Inamova M., Ortiqova K., Mirzaakbarov A.	
Enhancing cotton ginning processing method for better fibre quality	12
Kamalova I., Inoyatova M., Rustamova S., Madaliyeva M.	
Creating a patterned decorative landscape using knitted shear waste on the surface of the paint product	16
Inoyatova M., Ergasheva Sh., Kamalova I., Toshpo'latov M.	
State of development of fiber products – cleaning, combing techniques and technologies	21
Vakhobova N., Nigmatova F., Kozhabergenova K.	
Study of clothing requirements for children with cerebral palsy	30
Mukhametshina E., Muradov M.	
Analysis of the improvement of pneumatic outlets in the pneumatic transport system	37
Otamirzayev A.	
Innovative solutions for dust control in cotton gining enterprises	45
Muradov M., Khuramova Kh.	
Studying the types and their composition of pollutant mixtures containing cotton seeds	50
Mukhamedjanova S.	
Modernized sewing machine bobbin cap hook thread tension regulator	53
Ruzmetov R., Kuliyeu T., Tuychiev T.	
Study of effect of drying agent component on cleaning efficiency.	57
Kuldashov G., Nabiev D.	
Optoelectronic devices for information transmission over short distances	65
Kuliev T., Abbasov I., F.Egamberdiev.	
Improving the elastic mass of fiber on the surface of the saw cylinder in fiber cleaning equipment using an additional device	73
Yusupov A., Muminov M., Iskandarova N., Shin I.	

On the influence of the wear resistance of grate bars on the technological gap between them in fiber separating machines **80**

Kuliev T., Jumabaev G., Jumaniyazov Q.

Theoretical study of fiber behavior in a new structured elongation pair **86**

GROWING, STORAGE, PROCESSING AND AGRICULTURAL PRODUCTS AND FOOD TECHNOLOGIES

Meliboyev M., Ergashev O., Qurbonov U.

Technology of freeze-drying of raw meat **96**

Davlyatov A., Khudaiberdiev A., Khamdamov A.

Physical-chemical indicators of plum oil obtained by the pressing method **102**

Tojibaev M., Khudaiberdiev A.

Development of an energy-saving technological system to improve the heat treatment stage of milk **109**

Turg'unov Sh., Mallabayev O.

Development of technology for the production of functional-oriented bread products **115**

Voqqosov Z., Khodzhiev M.

Description of proteins and poisons contained in flour produced from wheat grain produced in our republic **120**

CHEMICAL TECHNOLOGIES

Choriev I., Turaev Kh., Normurodov B.

Determination of the inhibitory efficiency of the inhibitor synthesized based on maleic anhydride by the electrochemical method **126**

Muqumova G., Turayev X., Mo'minova Sh., Kasimov Sh., Karimova N.

Spectroscopic analysis of a sorbent based on urea, formalin, and succinic acid and its complexes with ions of Cu(II), Zn(II), Ni(II) **131**

Babakhanova Kh., Abdukhalilova M.

Analysis of the composition of the fountain solution for offset printing **138**

Babakhanova Kh., Ravshanov S., Saodatov A., Saidova D.

Development of the polygraphic industry in the conditions of independence **144**

Tursunqulov J., Kutlimurotova N., Jalilov F., Rahimov S.

Determination zirconium with the solution of 1-(2-hydroxy-1-naphthoyazo)-2-naphthol-4-sulfate **151**

Allamurtova A., Tanatarov O., Sharipova A., Abdikamalova A., Kuldasheva Sh.

Synthesis of acrylamide copolymers with improved viscosity characteristics **156**

Amanova N., Turaev Kh., Alikulov R., Khaitov B., Eshdavlatov E., Makhmudova Y.	
Research physical and mechanical properties and durability of sulfur concrete	165

MECHANICS AND ENGINEERING

Abdullaev E., Zakirov V.	
Using parallel service techniques to control system load	170
Djuraev R., Kayumov U., Pardaeva Sh.	
Improving the design of water spray nozzles in cooling towers	178
Anvarjanov A., Kozokov S., Muradov R.	
Analysis of research on changing the surface of the grid in a device for cleaning cotton from fine impurities	185
Mahmudjonov M.	
Mathematical algorithm for predicting the calibration interval and metrological accuracy of gas analyzers based on international recommendations ILAC-G24:2022/OIML D 10:2022 (E)	192
Kulmuradov D.	
Evaluation of the technical condition of the engine using the analysis of the composition of gases used in internal combustion engines	197
Kiryigitov Kh., Taylakov A.	
Production wastewater treatment technologies (On the example of Ultramarine pigment production enterprise).	203
Abdullayev R.	
Improving the quality of gining on products.	208
Abdullayev R.	
Problems and solutions to the quality of the gining process in Uzbekistan.	212
Yusupov D., Avazov B.	
Influence of various mechanical impurities in transformer oils on electric and magnetic fields	216
Kharamonov M.	
Prospects for improving product quality in textile industry enterprises based on quality policy systems	223
Kharamonov M., Kosimov A.	
Problems and solutions to the quality of the gining process in Uzbekistan.	230
Mamahonov A., Abdusattarov B.	
Development of simple experimental methods for determining the coefficient of sliding and rolling friction.	237

Aliyev E., Mamahonov A.	
Development of a new rotary feeder design and based flow parameters for a seed feeder device	249
Ibrokhimova D., Akhmedov K., Mirzaumidov A.	
Theoretical analysis of the separation of fine dirt from cotton.	260
Razikov R., Abdazimov Sh., Saidov D., Amirov M.	
Causes of floods and floods and their railway and economy influence on construction.	266
Djurayev A., Nizomov T.	
Analysis of dependence on the parameters of the angles and loadings of the conveyor shaft and the drum set with a curved pile after cleaning cotton from small impurities	272
ADVANCED PEDAGOGICAL TECHNOLOGIES IN EDUCATION	
Jabbarov S.	
Introduction interdisciplinary nature to higher education institutions.	276
Tuychibaev H.	
Analysis of use of sorting algorithms in data processing.	280
Kuziev A.	
Methodology for the development of a low cargo network.	289
Niyozova O., Turayev Kh., Jumayeva Z.	
Analysis of atmospheric air of Surkhondaryo region using physico-chemical methods.	298
Isokova A.	
Analysis of methods and algorithms of creation of multimedia electronic textbooks.	307
ECONOMICAL SCIENCES	
Rashidov R., Mirjalolova M.	
Regulations of the regional development of small business.	315
Israilov R.	
Mechanism for assessment of factors affecting the development of small business subjects.	325
Yuldasheva N.	
Prospects of transition to green economy.	334
Malikova G.	
Analysis of defects and solutions in investment activity in commercial banks.	346