Meeting 8

Avicena Yosanta Muchtar 24/536710/PA/22764

14 November 2024

https://github.com/TheKurusUGM/Praktikum-Pemograman-UGM-/tree/main

1 Question 1:

Question:
Sort the data in descending by:
- NISN
- Value
using insertion sort, selection sort, and bubble sort.

Answer:

```cpp
#include <iostream>
#include <vector>
#include <algorithm>

struct Student {
    long long NISN;
    std::string Nama;
    int Value;
};

void displayData(const std::vector<Student>& students) {
    for (const auto& student : students) {
        std::cout << student.NISN << " " << student.Nama << " "
        ↪  << student.Value << std::endl;
    }
}

void insertionSortByNISN(std::vector<Student>& students) {
    for (size_t i = 1; i < students.size(); i++) {
        Student key = students[i];
        int j = i - 1;
        while (j >= 0 && students[j].NISN < key.NISN) {
            students[j + 1] = students[j];
```

```
            j--;
        }
        students[j + 1] = key;
    }
}

void selectionSortByValue(std::vector<Student>& students) {
    for (size_t i = 0; i < students.size() - 1; i++) {
        size_t maxIdx = i;
        for (size_t j = i + 1; j < students.size(); j++) {
            if (students[j].Value > students[maxIdx].Value) {
                maxIdx = j;
            }
        }
        std::swap(students[i], students[maxIdx]);
    }
}

void bubbleSortByNISN(std::vector<Student>& students) {
    bool swapped;
    for (size_t i = 0; i < students.size() - 1; i++) {
        swapped = false;
        for (size_t j = 0; j < students.size() - i - 1; j++) {
            if (students[j].NISN < students[j + 1].NISN) {
                std::swap(students[j], students[j + 1]);
                swapped = true;
            }
        }
        if (!swapped) break;
    }
}

void bubbleSortByValue(std::vector<Student>& students) {
    bool swapped;
    for (size_t i = 0; i < students.size() - 1; i++) {
        swapped = false;
        for (size_t j = 0; j < students.size() - i - 1; j++) {
            if (students[j].Value < students[j + 1].Value) {
                std::swap(students[j], students[j + 1]);
                swapped = true;
            }
        }
        if (!swapped) break;
    }
}

```

```cpp
int main() {
    std::vector<Student> students = {
        {9960312699, "Handi Ramadhan", 90},
        {9963959682, "Rio Alfandra", 55},
        {9950310962, "Ronaldo Valentino Uneputty", 80},
        {9970272750, "Achmad Yaumil Fadjri R.", 60},
        {9970293945, "Alivia Rahma Pramesti", 70},
        {9952382180, "Ari Lutfianto", 65},
        {9965653989, "Arief Budiman", 60}
    };

    std::cout << "Original Data:\n";
    displayData(students);

    insertionSortByNISN(students);
    std::cout << "\nSorted by NISN (Descending) using Insertion
    ↪  Sort:\n";
    displayData(students);

    selectionSortByValue(students);
    std::cout << "\nSorted by Value (Descending) using Selection
    ↪  Sort:\n";
    displayData(students);

    students = {
        {9960312699, "Handi Ramadhan", 90},
        {9963959682, "Rio Alfandra", 55},
        {9950310962, "Ronaldo Valentino Uneputty", 80},
        {9970272750, "Achmad Yaumil Fadjri R.", 60},
        {9970293945, "Alivia Rahma Pramesti", 70},
        {9952382180, "Ari Lutfianto", 65},
        {9965653989, "Arief Budiman", 60}
    };

    bubbleSortByNISN(students);
    std::cout << "\nSorted by NISN (Descending) using Bubble
    ↪  Sort:\n";
    displayData(students);

    students = {
        {9960312699, "Handi Ramadhan", 90},
        {9963959682, "Rio Alfandra", 55},
        {9950310962, "Ronaldo Valentino Uneputty", 80},
        {9970272750, "Achmad Yaumil Fadjri R.", 60},
        {9970293945, "Alivia Rahma Pramesti", 70},
        {9952382180, "Ari Lutfianto", 65},
```

```
        {9965653989, "Arief Budiman", 60}
    };

    bubbleSortByValue(students);
    std::cout << "\nSorted by Value (Descending) using Bubble
    ↪  Sort:\n";
    displayData(students);

    return 0;
}
```

2 Explanation

```
struct Student {
    long long NISN;
    std::string Nama;
    int Value;
};
```

The struct Student defines the data structure for each student's information.

```
void displayData(const std::vector<Student>& students) {
    for (const auto& student : students) {
        std::cout << student.NISN << " " << student.Nama << " "
        ↪  << student.Value << std::endl;
    }
}
```

The displayData function displays each student's information from the vector.

```
void insertionSortByNISN(std::vector<Student>& students) {
    for (size_t i = 1; i < students.size(); i++) {
        Student key = students[i];
        int j = i - 1;
        while (j >= 0 && students[j].NISN < key.NISN) {
            students[j + 1] = students[j];
            j--;
        }
        students[j + 1] = key;
    }
}
```

This function sorts students in descending order based on NISN.
For each element in the list, it finds the correct position by constantly changing elements until NISN is in descending order.

```cpp
void selectionSortByValue(std::vector<Student>& students) {
    for (size_t i = 0; i < students.size() - 1; i++) {
        size_t maxIdx = i;
        for (size_t j = i + 1; j < students.size(); j++) {
            if (students[j].Value > students[maxIdx].Value) {
                maxIdx = j;
            }
        }
        std::swap(students[i], students[maxIdx]);
    }
}
```

This function sorts students by Value in descending order.
It finds the maximum value in the remaining unsorted part of the list and swaps it with the current position.

bubbleSortByNISN and bubbleSortByValue are two separate bubble sorts, one for NISN and one for Value.
Bubble Sort repeatedly swaps elements if they are out of order until the entire list is sorted.

```cpp
int main() {
    std::vector<Student> students = {
        {9960312699, "Handi Ramadhan", 90},
        {9963959682, "Rio Alfandra", 55},
        // ... other students ...
    };
    // Sort and display results for each sort
}
```

Initializes the list of students.
- Calls each sorting function on a copy of students to sort it by NISN and Value in descending order using different algorithms.
- After each sort, it displays the sorted list.

3 Question 2

Question:
Look for data that has NISN 9950310962, then show the value using the binary search.

```cpp
#include <iostream>
#include <vector>
#include <algorithm>

```

```cpp
struct Student {
    long long NISN;
    std::string Nama;
    int Value;
};

// Binary search function
int binarySearch(const std::vector<Student>& students, long long targetNISN) {
    int left = 0;
    int right = students.size() - 1;
    while (left <= right) {
        int mid = left + (right - left) / 2;
        if (students[mid].NISN == targetNISN) {
            return students[mid].Value; // Return the Value if found
        } else if (students[mid].NISN < targetNISN) {
            left = mid + 1;
        } else {
            right = mid - 1;
        }
    }
    return -1; // Return -1 if NISN is not found
}

int main() {
    std::vector<Student> students = {
        {9960312699, "Handi Ramadhan", 90},
        {9963959682, "Rio Alfandra", 55},
        {9950310962, "Ronaldo Valentino Uneputty", 80},
        {9970272750, "Achmad Yaumil Fadjri R.", 60},
        {9970293945, "Alivia Rahma Pramesti", 70},
        {9952382180, "Ari Lutfianto", 65},
        {9965653989, "Arief Budiman", 60}
    };

    // Sort the students by NISN (ascending) to prepare for binary search
    std::sort(students.begin(), students.end(), [](const Student& a, const Student& b) {
        return a.NISN < b.NISN;
    });

    long long targetNISN = 9950310962;
    int value = binarySearch(students, targetNISN);

```

```
    if (value != -1) {
        std::cout << "The value for NISN " << targetNISN << "
        ↪  is: " << value << std::endl;
    } else {
        std::cout << "NISN " << targetNISN << " not found." <<
        ↪  std::endl;
    }

    return 0;
}
```

Binary Search Function (binarySearch):

- This function performs binary search on the sorted vector of Student structures.
- If the targetNISN matches a student's NISN, it returns the Value.
- If the NISN is not found, it returns -1.

Sorting by NISN:

- Before applying binary search, the code sort the students vector by NISN in ascending order using std::sort with a lambda function.
- The main function searches for NISN 9950310962 and outputs the associated Value if found. Otherwise, it prints "not found."

4 Question 3

Question:
Change the name of the data that has a value of 60 to Joko. Take advantage of sequential search methods.

```
#include <iostream>
#include <vector>

struct Student {
    long long NISN;
    std::string Nama;
    int Value;
};

void displayData(const std::vector<Student>& students) {
    for (const auto& student : students) {
        std::cout << student.NISN << " " << student.Nama << " "
        ↪  << student.Value << std::endl;
    }
}


```

```cpp
void updateNameByValue(std::vector<Student>& students, int targetValue, const std::string& newName) {
    for (auto& student : students) {
        if (student.Value == targetValue) {
            student.Nama = newName;
        }
    }
}

int main() {
    std::vector<Student> students = {
        {9960312699, "Handi Ramadhan", 90},
        {9963959682, "Rio Alfandra", 55},
        {9950310962, "Ronaldo Valentino Uneputty", 80},
        {9970272750, "Achmad Yaumil Fadjri R.", 60},
        {9970293945, "Alivia Rahma Pramesti", 70},
        {9952382180, "Ari Lutfianto", 65},
        {9965653989, "Arief Budiman", 60}
    };

    std::cout << "Original Data:\n";
    displayData(students);


    updateNameByValue(students, 60, "Joko");

    std::cout << "\nData after updating names with Value 60 to 'Joko':\n";
    displayData(students);

    return 0;
}
```

5 Explanation

```cpp
struct Student {
    long long NISN;
    std::string Nama;
    int Value;
};
```

Defines the Student struct with NISN, Nama, and Value fields to represent each student's data.

```cpp
void updateNameByValue(std::vector<Student>& students, int
↪  targetValue, const std::string& newName) {
    for (auto& student : students) {
        if (student.Value == targetValue) {
            student.Nama = newName;
        }
    }
}
```

This function searches for students with a Value of 60 and updates their Nama to "Joko".

```cpp
int main() {
    std::vector<Student> students = {
        {9960312699, "Handi Ramadhan", 90},
        {9963959682, "Rio Alfandra", 55},
        {9950310962, "Ronaldo Valentino Uneputty", 80},
        {9970272750, "Achmad Yaumil Fadjri R.", 60},
        {9970293945, "Alivia Rahma Pramesti", 70},
        {9952382180, "Ari Lutfianto", 65},
        {9965653989, "Arief Budiman", 60}
    };

    updateNameByValue(students, 60, "Joko");

    displayData(students);
    return 0;
}
```

Initializes the students vector with the given data.
Calls updateNameByValue to change the name of students with Value of 60 to "Joko".
Displays data before and after the update.