**SPRAWOZDANIE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Przedmiot** | Wprowadzenie do Informatyki | **Zadanie** | 7.3 |
| **Autor** | Andrii Godlevskyi | **Grupa** | WCY23KY1S1 |
| **Temat** | Podstawowe metody sortowania | | |

1.Treść zadania

Należy napisać program, który umożliwi:

Zaimplementuj algorytm sortowania bąbelkowego (Insertion Sort)

Metoda realizacji

Tworzymy tablice i funkcje wypełnienia ją losowymi elementami zanim robimy funkcje sortowania i wypisania tablicy, dalej wyprowadzamy wynik na ekran.

1. Założenia / ograniczenia dotyczące danych:

1. Dane wejściowe

n - ilość elementów tablicy do posortowania – wczytywana z klawiatury

1. Dane wyjściowe

Elementy tablic przed i po sortowaniu z nagłówkami (po 10 w wierszu, a każda liczba zajmuje tyle samo miejsca) - wyświetlone na ekranie *(z opisami).*

1. Realizacja
2. Algorytm

Изображение выглядит как текст, рукописный текст, бумага, блокнот

Автоматически созданное описание



1. Kod zródłowy

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

//Funkcja sortowania

void insertionSort(int arr[], int n) {

int i, key, j;

for (i = 1; i < n; i++) {

key = arr[i]; // Przechowuje bieżący element

j = i - 1;

while (j >= 0 && arr[j] > key) {

arr[j + 1] = arr[j];

j = j - 1;

}

// Wstawia 'key' na właściwe miejsce

arr[j + 1] = key;

}

}

// Funkcja wypisania tablicy

void printtab(int tab[], int n) {

for (int i = 0; i < n; i++) {

printf(" %d ", tab[i]);

if ((i + 1) % 10 == 0) {

printf("\n");

}

}

printf("\n");

}

// Funnkcja wypelniena tablicy losowymi elementami

void wyptab(int arr[], int N, int a, int b) {

for (int i = 0; i < N; i++) {

arr[i] = rand()%(b-a+1)+a;

}

}

int main()

{

//deklaracja zmiennych

int arr[1000];

int n, a, b;

int N = sizeof(arr) / sizeof(arr[0]);

//wprowadzenie wartosci zmiennych

printf("Wpisz rozmiar tablicy: ");

scanf("%d", &N);

printf("wpisz przedzial [a, b]: ");

scanf("%d %d", &a, &b);

srand(time(0));//genetator losowych liczb

//wywolanie funkcji

wyptab(arr, N, a, b);

//wypisanie poczatkowej tabeli

printf("Nieposortowana tablica: \n");

printtab(arr, N);

//wywolanie sortowania

insertionSort(arr, N);

//wypisanie posortowanej tabeli

printf("Posortowana tablica: \n");

printtab(arr, N);

return 0;

}

1. Dane wejściowe /wyjściowe

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

1. Złożoność obliczeniowa algorytmu

O(1)