**SPRAWOZDANIE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Przedmiot** | Algorytmy i struktury danych | **Zadanie** | 5\_01 |
| **Autor** | Kamil Borkowski | **Grupa** | WCY22IY1S1 |
| **Temat** | Wieża Hanoi | | |

1. Treść

Zaimplementuj algorytm realizujący problem wieży Hanoi

Metoda realizacji

Utworzenie trzech stosów reprezentujących wieże na krążki, przekładanie krążków według schematu AB,AC,BC – wykonywanie jedynych legalnych ruchów między danymi wieżami aż do ułożenia całej wieży na wieży C

* 1. Założenia / ograniczenia dotyczące danych:
     1. Dane wejściowe *(opis)*

Ilość krążków – n– wczytywana z klawiatury

* + 1. Dane wyjściowe *(opis)*

Informacje o kolejności przekładania krążków – wyświetlone na ekranie i w pliku .txt

1. Realizacja
   1. Algorytm *(lista kroków lub schemat blokowy)*

Deklaracja struktury wieży

Utworzenie wskaźnikowo trzech wież

Uzupełnienie pierwszej wieży krążkami

Przenoszenie krążków według schematu z metody realizacji

Wypisywanie na ekranie i w pliku .txt wszystkich zmian na wieżach

Zakończenie programu po ułożeniu całej wieży C

* 1. Kod źródłowy

//================================================

//Zadanie 5\_01 jezyk C++

//Wieza Hanoi

//WCY21IY1S1 Borkowski Kamil

//================================================

#include <iostream>

#include <cstdlib>

#include <fstream>

using namespace std;

struct wieza

{

int k;

wieza\* prev;

};

void printTowerToFile(wieza\* head, ofstream& file)

{

wieza\* current = head;

while (current != NULL)

{

file << current->k << '\t';

cout << current->k << '\t';

current = current->prev;

}

file << endl;

cout<< endl;

}

void przenies\_krazek(wieza\*& OD, wieza\*& DO, wieza\*& Ahead, wieza\*& Bhead, wieza\*& Chead, ofstream& file)

{

wieza\* temp = OD;

OD = OD->prev;

temp->prev = DO;

DO = temp;

// Dodatkowy kod do wypisywania stanu wiez po przeniesieniu krazka

cout << "Wieza A" << endl;

file << "Wieza A" << endl;

printTowerToFile(Ahead, file);

cout << "Wieza B" << endl;

file << "Wieza B" << endl;

printTowerToFile(Bhead, file);

cout << "Wieza C" << endl;

file << "Wieza C" << endl;

printTowerToFile(Chead, file);

cout << "-----------------------------------------" << endl;

file << "-----------------------------------------" << endl;

}

void hanoi(int n, wieza\*& OD, wieza\*& PRZEZ, wieza\*& DO, wieza\*& Ahead, wieza\*& Bhead, wieza\*& Chead, ofstream& file)

{

if (n == 1)

{

przenies\_krazek(OD, DO, Ahead, Bhead, Chead, file);

return;

}

hanoi(n - 1, OD, DO, PRZEZ, Ahead, Bhead, Chead, file);

przenies\_krazek(OD, DO, Ahead, Bhead, Chead, file);

hanoi(n - 1, PRZEZ, OD, DO, Ahead, Bhead, Chead, file);

}

int main()

{

int n, i;

cout << "Podaj ilosc krazkow:" << endl;

cin >> n;

ofstream wy("wy.txt");

wieza\* Ahead = NULL; // deklaracja 3 wiez

wieza\* Bhead = NULL;

wieza\* Chead = NULL;

i = n; // umieszczanie n krazkow na stosie A

while (i > 0)

{

wieza\* krazek = (wieza\*)malloc(sizeof(wieza));

krazek->k = i;

i--;

krazek->prev = Ahead;

Ahead = krazek;

}

cout << "Poczatkowy stan wiez z krazkami" << endl;

wy << "Poczatkowy stan wiez z krazkami" << endl;

cout << "Wieza A" << endl;

wy << "Wieza A" << endl;

printTowerToFile(Ahead, wy);

cout << "Wieza B" << endl;

wy << "Wieza B" << endl;

printTowerToFile(Bhead, wy);

cout << "Wieza C" << endl;

wy << "Wieza C" << endl;

printTowerToFile(Chead, wy);

wy << "===============================" << endl;

cout << "===============================" << endl;

hanoi(n, Ahead, Bhead, Chead, Ahead, Bhead, Chead, wy);

cout << "Koncowy stan wiez z krazkami" << endl;

wy << "Koncowy stan wiez z krazkami" << endl;

cout << "Wieza A" << endl;

wy << "Wieza A" << endl;

printTowerToFile(Ahead, wy);

cout << "Wieza B" << endl;

wy << "Wieza B" << endl;

printTowerToFile(Bhead, wy);

cout << "Wieza C" << endl;

wy << "Wieza C" << endl;

printTowerToFile(Chead, wy);

wy.close();

return 0;

}

* 1. Dane wejściowe *(ekran uruchomienia programu)*

Obraz zawierający zrzut ekranu, tekst

Opis wygenerowany automatycznie

* 1. Dane wyjściowe *(ekran uruchomienia programu)*

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

1. Wnioski *(złożoność asymptotyczna algorytmu przy użyciu notacji* ***O lub innej i inne wnioski)***

Złożoność obliczeniowa programu wynosi: O(2^n), zależna od ilości krążków .