**SPRAWOZDANIE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Przedmiot** | Algorytmy i struktury danych | **Zadanie** | 3\_1 |
| **Autor** | Krzysztof Chrapowicz | **Grupa** | WCY22IY1S1 |
| **Temat** | Sito Eratostenesa | | |

1. Treść zadania

Zaimplementuj algorytm wyznaczania liczb pierwszych z zadanego przedziału [2,n]. (Użyj listy jednokierunkowej zaimplementowanej wskaźnikowo)

* 1. Metoda realizacji

Algorytm opiera się na wykorzystaniu listy jednokierunkowej do przechowywania liczb od 2 do n, a następnie eliminuje wielokrotności kolejnych liczb pierwszych.

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

Liczba n oznaczająca górny zakres ciągu liczb pierwszych

* + 1. Dane wyjściowe *(opis)*

Liczby pierwsze od 2 do n.

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

Inicjalizacja listy:

Tworzy się dynamiczną listę jednokierunkową, w której przechowywane są liczby od 2 do n.

Eliminacja wielokrotności liczb pierwszych:

Począwszy od 2, algorytm przechodzi przez listę i eliminuje wielokrotności aktualnie rozważanej liczby pierwszej.

Iteracyjnie zwiększa się wartość p (aktualnej liczby pierwszej), eliminuje wielokrotności tej liczby i przechodzi do następnej liczby pierwszej, aż do momentu, gdy kwadrat aktualnej liczby pierwszej przekracza n.

Wyświetlenie liczb pierwszych:

Po zakończeniu eliminacji, algorytm wyświetla na ekranie pozostałe liczby w liście, które są liczbami pierwszymi w zadanym zakresie.

* 1. Kod źródłowy

#include <iostream>

#include <cstdlib>

using namespace std;

struct Node

{

int value;

Node\* next;

};

void erastotenes(int n)

{

Node\* head = NULL;

Node\* tail = NULL;

// Wypelnienie listy liczbami

for(int i = 2; i <= n; ++i)

{

Node\* new\_node = (Node\*)malloc(sizeof(Node));

new\_node->value = i;

new\_node->next = NULL;

if(head == NULL)

{

head = new\_node;

tail = new\_node;

}

else

{

tail->next = new\_node;

tail = new\_node;

}

}

int p = 2;

while (p \* p <= n)

{

Node\* current = head;

Node\* prev = NULL;

while(current != NULL)

{

int num = current->value;

if(num == p)

{

prev = current;

current = current->next;

}

//sprawdzenie czy liczba mimo nie bycia rownej p jest powiedzielna przez nia, i usuniecie jej gdy jest

else if(num % p == 0)

{

if(prev != NULL)

{

prev->next = current->next;

}

else

{

head = current->next;

}

free(current);

current = prev->next;

}

else

{

prev = current;

current = current->next;

}

}

p += 1;

}

//zwolnienie pammieci

Node\* current = head;

while(current != NULL)

{

cout << current->value << " ";

Node\* temp = current;

current = current->next;

free(temp);

}

}

int main()

{

int n;

cout << "Podaj n: ";

cin >> n;

cout << "Liczby pierwsze: ";

erastotenes(n);

return 0;

}

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

Obraz zawierający zrzut ekranu, tekst, czarne, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

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

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

1. Wnioski

Złożoność czasowa wynosi O(n log(log n)).