# Dijkstra-Parte Secventiala

1. **Datele si codul pentru varianta secventiala trebuie puse intr-un repository pe** [**https://github.com/RaduNDL/Dijkstra.git**](https://github.com/RaduNDL/Dijkstra.git) **In repository trebuie sa existe un gdoc in care sunt specificate:**
2. cerintele/tema proiectului:

Drumuri Minime: Algoritmul Dijkstra

* Descriere

Acest program implementeaza algoritmul Dijkstra pentru gasirea drumurilor minime intr-un graf, utilizand limbajul de programare C++. Algoritmul Dijkstra este utilizat pentru a calcula cel mai scurt drum de la un nod sursa catre toate celelalte noduri dintr-un graf ponderat. Implementarea prezenta este secventiala si utilizeaza structuri de date eficiente pentru a gestiona graful si coada de prioritati.

* Caracteristici

- Algoritm: Dijkstra

- Limbaj de programare: C++

- Functionalitate: Calculul drumurilor minime intr-un graf ponderat

- Iesire: Salvarea distantelor minime intr-un fisier text

2) info despre masinii pe care ati rulat codul:

Processor AMD Ryzen 9 6900HX with Radeon Graphics 3.30 GHz

Installed RAM 16.0 GB (15.3 GB usable)

GPU: NVIDIA GeForce RTX 3070 Ti Laptop 8GB

System type 64-bit operating system, x64-based processor

3)rezultatele experimentale, adica timpii de rulare.

* int V = 10000; int E = 100000;

Total Execution Time: 0.468141 seconds

* int V = 100000; int E = 1000000;

Total Execution Time: 5.92512 seconds

* int V = 1000000; int E = 10000000;

Total Execution Time: 73.7692 seconds

* int V = 900000; int E = 1000000;

Total Execution Time: 11.2368 seconds

* int V = 999999; int E = 1000000;

Total Execution Time: 11.4396 seconds

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| V | **E** | **Graph Generation (s)** | **Graph Reading (s)** | **Dijkstra (s)** |
| 100000 | 1000000 | 8.7737 | 5.40985 | 0.574029 |
| 10000 | 100000 | 0.903871 | 0.510063 | 0.0570666 |
| 1000000 | 10000000 | 93.5132 | 58.7466 | 6.58402 |
| 900000 | 1000000 | 9.37112 | 5.57059 | 4.22038 |
| 999999 | 1000000 | 9.52905 | 5.26958 | 4.30565 |

1. **S11. Varianta paralelă #1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| V | **E** | **Graph Generation (s)** | **Graph Reading (s)** | **Dijkstra (s)** |
| 100000 | 1000000 | 5.555909 | 5.525262 | 0.087592 |
| 10000 | 100000 | 0.564045 | 0.572811 | 0.013690 |
| 1000000 | 10000000 | 53.969915 | 51.145300 | 0.716705 |
| 900000 | 1000000 | 5.470187 | 5.599010 | 0.576170 |
| 999999 | 1000000 | 5.506630 | 5.589575 | 0.663476 |

**Paralel vs secvential:**

|  |  |  |
| --- | --- | --- |
|  | **Dijkstra (s)Paralel** | **Graph Generation (s)Secvential** |
| 100000 | 0.08759 | 5.555909 |
| 10000 | 0.01369 | 0.564045 |
| 1000000 | 0.71671 | 53.969915 |
| 900000 | 0.57617 | 5.470187 |
| 999999 | 0.66348 | 5.50663 |

