Vilniaus Vytauto Didžiojo Gimnazija

Deividas Smolskis IIA

Rikiavimo burbuliuko metodo algoritmas C++, Python, JavaScript ir C# programavimo kalbose.

TURINYS

1. ĮVADAS
2. TIKSLAS
3. HIPOTEZĖ
4. VEIKLŲ PLANAS
5. ŠALTINIŲ ANALIZĖ
6. TIRIAMOJI DALIS
7. REZULTATAS
8. IŠVADOS
9. LITERATŪROS ŠALTINIŲ SĄRAŠAS

ĮVADAS

Šią temą pasirinkau tikėdamasis nustatyti ir parodyti programavimo kalbą, kurioje „Bubble Sort“ algoritmas veikia optimaliausiai – greičiausiai sukompiliuojamas ir pavaizduojamas norimas rezultatas. Tema yra ypatingai aktuali šiandieniniame pasaulyje, nes dėl šio algoritmo galime greitai ir ypatingai lengvai surikiuoti duomenis eilės tvarka. Kadangi pačio algoritmo optimizacija nėra tobula (ypač ties laiku reikalingu kompiuterio CPU atlikti tam tikrą procesą ), privaloma rasti kalbą, kurioje jis veikia optimaliausiai.

TIKSLAS

Tyrimu bus siekiama nustatyti optimaliausią programavimo kalbą rikiavimo burbuliuko algoritmui, bus pamatuojamas įvairiais rodikliais, pavyzdžiui, algoritmo atminties matavimu, algoritmo įvykdymo laiku, tyrimas bus įgyvendinamas parašius ir išbandžius rikiavimo burbuliuko algoritmą, kadangi tai vienas iš daugiausiai naudojamų algoritmų.

HIPOTEZĖ

Atlikus ir įgyvendinus tyrimą tikiuosi pamatyti C++ kaip optimaliausią ir greičiausią programavimo kalbą įvykdžiusią burbuliuko rikiavimo algoritmą, jei tyrimas pakryptų nenumatyta linkme, peržiūrėsiu kodą ir bandysiu jį optimizuoti.

VEIKLŲ PLANAS

I-oje gimnazijos klasėje pradėsiu savo tyrimą - palyginsiu rikiavimo burbuliuko metodo algoritmą C++, JavaScript programavimo kalbose, bandysiu jį teisingai įgyvendinti ir suprasti. II-oje gimnazijos klasėje baigsiu savo tyrimą – galiausiai šį algoritmą palyginsiu Python, C# programavimo kalbose. Tyrimui įvykdyti pasinaudosiu kompiuteriu ir įvairia programine įranga (t.y Visual Studio, Visual Studio Code). Darbo įgyvendinimas užtruks 2 mokslo metus, per tą laiką gilinsiuosi, kaip algoritmai matuojami akademiniuose moksluose ir tikiuosi perprasti rikiavimo burbuliuko metodo algoritmą ir tiriamąjį darbą pristatyti gimnazijos bendruomenės nariams.

ŠALTINIŲ ANALIZĖ

Burbuliuko rikiavimo algoritmą lyginant su kitais panašiais ir labiau optimizuotais algoritmais, matome, jog jis lėtas ir neefektyvus su dideliu elementų kiekiu (<https://www.geeksforgeeks.org/bubble-sort/>). Taip pat, šis algoritmas pasižymi labai prasta modernių procesorių optimizacija – jis reikalauja daugiau nei dvigubai procesoriaus išteklių lyginant su šio algoritmo alternatyvomis, kaip įterpimo rikiavimo algoritmas (<https://www.baeldung.com/cs/bubble-sort-time-complexity>). Tačiau tai nereiškia, kad šis algoritmas neturi prasmės – jis lengvai suprantamas ir įgyvendinamas, nereikalingi papildomi kompiuterio atminties ištekliai, lengvai ir įvairiai optimizuojamas bei, be abejo, yra stabilus (<https://www.simplilearn.com/tutorials/data-structure-tutorial/bubble-sort-algorithm>).

TIRIAMOJI DALIS

I-oje klasėje pradėjau darbą – ištyriau „bubble sort“ algoritmą C++, JavaScript kalbose. Algoritmui parašyti patirties neturėjau, todėl naudojausi <https://www.geeksforgeeks.org/bubble-sort/>.