

Oblig 1 ADS101

Adam Aske

28. september 2021

Innhold

1	Introduksjon	2
2	Selection Sort	2
2.1	Selection Sort Results	2
3	Std::Sort	2
4	Std::Sort Resultat	3

1 Introduksjon

Her bruker jeg algoritmen Selection Sort og C++ sin innebygde funksjon Sort til å sortere arrayer av forskjellige størrelser. Chrono tar tiden på hvor lang tid det tar for arrayer av forskjellige størrelser. Målingene er gjennomsnittet av algoritmene som har blitt kjørt 10 ganger per N.

2 Selection Sort

Selection sort algoritmen på int arrayer av forskjellige størrelser. Funksjonen tar inn en array og en størrelse. Først blir arrayen fylt med tilfeldige tall.

Listing 1: main.cpp

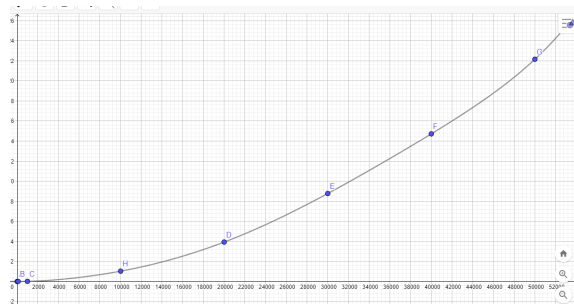
```
template<typename T, size_t N>
void SelectionSort(T (&arr)[N], std::ofstream &file){

    for (int i = 0; i < N; i++){
        arr[i] = rand();
    }

    auto start = std::chrono::high_resolution_clock::now();
    //1. Selection sort
    for(auto i = 0; i < N-1; i++){
        for(auto j = i +1; j < N; j++){
            if(arr[j] > arr[i]){
                std::swap(arr[j], arr[i]);
            }
        }
    }

    //Get the end
    auto end = std::chrono::high_resolution_clock::now();
    std::chrono::duration<double> totalTime = end-start;
    std::chrono::nanoseconds totalTimeNano =
        std::chrono::duration_cast
            <std::chrono::nanoseconds>(totalTime);
    totalTime += totalTimeNano;
    int time = std::chrono::duration_cast<std::chrono
        ::nanoseconds>(end-start).count();
    std::cout << "With_" << N << "_elements, it took_"
        << totalTime.count()
        << "_nanoseconds to sort
        them doing the selection sort algorithm."
        << std::endl;
    file << totalTime.count() << std::endl;
    selectionTimes.push_back(totalTime.count());
}
```

2.1 Selection Sort Results



Figur 1: X-Aksen er antall elementer i arrayet og Y-Aksen er tiden det tok for å sortere elementene etter størrelse

3 Std::Sort

Std::Sort funksjonen brukes på de samme arrayene. Funksjonen tar inn en array og en størrelse. Først blir arrayen fylt med tilfeldige tall.

Listing 2: main.cpp

```
template<typename T, size_t N>
void StdSort(T (&arr)[N], std::ofstream &file){
    //2. Std::sort
    auto start = std::chrono::high_resolution_clock::now();

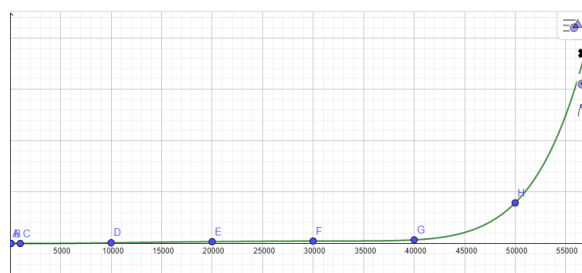
    std::sort(std::begin(arr), std::end(arr));

    auto end = std::chrono::high_resolution_clock::now();

    std::chrono::duration<double> totalTime = end-start;
    std::chrono::nanoseconds totalTimeNano =
        std::chrono::duration_cast
<std::chrono::nanoseconds>(totalTime);
    totalTime += totalTimeNano;
    int time = std::chrono::duration_cast
<std::chrono::nanoseconds>(end-start).count();
    std::cout << "With_" << N <<
        "_elements_it_took_" << totalTime.count()
        << "_nanoseconds_to_sort_them_using_std::sort."
    << std::endl;
    file << totalTime.count() << std::endl;
    stdSortTimes.push_back(totalTime.count());
}
```

4 Std::Sort Resultat

Resultat



Figur 2: X-Aksen er antall elementer i arrayet og Y-Aksen er tiden det tokk for å sortere elementene etter størrelse