Zestaw 5, Zadanie 2

Miłosz Bargieł

Obliczanie złożoności obliczeniowej pesymisrycznej, średniej oraz optymistycznej dla algorytmu sortowania Selection sort. Dominujaca operacja w tym algorytmie jest operacja porównania. W pierwszej iteracji wykonujemy $n-1$ porównań, w kolejnej $n-2$, nastepnie $n-3$ itd. Otrzymujemy nastepujaca sume: $(n-1)+(n-2)+(n-3)+...+3+2+1=\frac{n(n-1)}{2}$. Złożoność tej operacji jest zawsze równa $O(n^2)$ niezależnie od kolejności danych wejściowych. W celu ustalenia jaka jest złożoność optymistyczna, pesymistyczna oraz średnia porównam operacje `swap`.
Kod algorytmu:

```
template <typename T>
void sort(std::vector<T> &v) {
    int n = v.size();
    int min_idx;
    for (int i = 0; i < n-1; i++) {
        min_idx = i;
        for (int j = i+1; j < n; j++) {
            if (v.at(j) < v.at(min_idx))
                min_idx = j;
        }
        if (i != min_idx)
            std::swap(v.at(i), v.at(min_idx));
    }
}
```

1. Złożoność optymistyczna - scenariusz, w którym dane wejściowe sa już posortowane. Jak wyżej wspomniano wykonywane jest $O(n^2)$ porównań, lecz w przypadku gdy dane sa posortowane `i == min_idx` dla każdej iteracji wiec operacja swap nigdy nie jest wykonywana. Złożoność dla tej operacji wynosi zatem $O(1)$. Tak wiec optymistyczna złożoność to $O(n^2)$ porównań i $O(1)$ swapów.

2. Złożoność pesymityczna - scenariusz, w którym dane wejściowe sa ułożone w taki sposób, że dla każdego `i` warunek `i != min_idx` bedzie prawdziwy, wiec zostanie wykonane łacznie $n-1$ swapów. Tak wiec pesymistyczna złożoność to $O(n^2)$ porównań i $O(n)$ swapów.

3. Średnia złożoność - obliczana w nastepujacy sposób - wszystkie możliwe przypadki złożoności czasowej operacji podzielone przez liczbe przypadków.

Powyższy algorytm może wykonać $\boldsymbol{1+2+3+...+(n-2)+(n-1)}$ swapów. Liczba wszystkich możliwych scenariuszy wynosi $\boldsymbol{n}$. Tak wiec średnia złożoność to $\boldsymbol{O(n^2)}$ porównań i $\boldsymbol{\frac{n(n+1)}{2n} = \frac{n+1}{2} = O(n)}$ swapów. W celu sprawdzenia czy moje obliczenia sa poprawne dokonałem nastepujacej modyfikacji kodu:

```
template <typename T>
int sort(std::vector<T> &v) {
    int n = v.size();
    int min_idx;
    int swaps = 0;
    for (int i = 0; i < n-1; i++) {
        min_idx = i;
        for (int j = i+1; j < n; j++) {
            if (v.at(j) < v.at(min_idx))
                min_idx = j;
        }
        if (i != min_idx) {
            std::swap(v.at(i), v.at(min_idx));
            swaps++;
        }
    }
    return swaps;
}
```

Algorytm dokonujacy sortowania zlicza teraz ilość dokonanych zamian. W celu sprawdzenia obliczeń uruchomiłem program dajac mu do posortowania dane optymistyczne i n ciagów danych losowych zbudowanych z 1000 elementów.
Wyniki sa nastepujace:

(a) W wyniku działania algorytmu na 1000 losowo wygenerowanych ciagach liczb otrzymałem średnia równa: 992.437 co potwierdza, że średnia złożoność wynosi $\boldsymbol{O(n)}$ swapów.

(b) W wyniku działania algorytmu na ciagu 1000 posortowanych w kolejności malejacej elementów otrzymałem wynik 0 swapów, co potwierdza, że średnia złożoność wynosi $\boldsymbol{O(1)}$ swapów.

(c) W celu sprawdzenia złożoności pesymistycznej potrzebowałbym tak skonstruowanego zestawu danych, że przy każdej zamianie właściwego elementu z elementem który znajdował sie na indeksie `i` element ten trafiałby na niewłaściwe miejsce. Stworzenie takiego zestawu danych jest zadaniem nietrywialnym. Generujac losowe ciagi za pomoca algorytmu udało mi sie dojść do $\approx$ 999 swapów lecz nie osiagnałem tej liczby. W pliku `example.txt` znajduje sie ciag dla którego algorytm wykonał 994 swapy.