Analisis Kompleksitas Algoritma Selection Sort Iterative dan Recursive

Analisis Kompleksitas Algoritma

January 5, 2021

Disusun oleh

Ni Made Dwipadini Puspitarini	(1301194141)
Vena Erla Candrika	(1301194040)
Claudia Mei Serin Sitio	(1301190424)
Michael Putera Wardana	(1301194056)

1 Algoritma Selection Sort

Selection sort adalah metode pengurutan yang melakukan pencarian elemen yang terbesar (jika diurutkan mengecil) dan terkecil (jika diurutkan membesar) kemudian melakukan pertukaran terhadap elemen tersebut ke bagian paling depan elemen - elemen yang belum terurut.

1.1 Pseudo code

```
procedure selectionSortIterative(I/O : A: Array[1...n] of integer)
Kamus lokal:
Integer : i, j, min, temp
Algoritma :
for i=1 to n-1 do
        min = i
        for j=i+1 to n do
                if A[j] < A[min] then
                        min = j
        temp = A[min]
        A[min] = A[i]
        A[i] = temp
```

Figure 1: psuedocode iterative selection sort

```
procedure selectionSortRecursive(I/O: A: Array[1...n] of integer, Input : i: int)
Kamus lokal:
Integer : j, min, temp
Algoritma:
Min = i
if i+1 < n then
        for j=i+1 to n do
                if A[j] < A[min] then
                        min = j
        temp = A[min]
        A[min] = A[i]
        A[i] = temp
        selectionSortRecursive(A,i+1)
```

Figure 2: psuedocode recursive selection sort

1.2 Source code

```python
# Procedure recursive selection sort
def selectionSortRecursive(A, i):
    min = i
    n = len(A)
    if i + 1 < n:
        for j in range(i + 1, n):
            if A[j] < A[min]:
                min = j
        temp = A[min]
        A[min] = A[i]
        A[i] = temp
        selectionSortRecursive(A, i + 1)
```

```python
# Procedure iterative selection sort
def selectionSortIterative(A):
    for i in range(len(A) - 1):
        min = i
        for j in range(i + 1, len(A)):
            if A[j] < A[min]:
                min = j
        temp = A[min]
        A[min] = A[i]
        A[i] = temp
```

2 Worst Case dan Best Case

Melakukan pengujian dengan menggunakan input size 100 yang sudah terurut membesar dan mengecil. Data yang diujikan sebagai berikut :
$Bestcase = [0, 1, 2, \ldots, 98, 99]$
$Worstcase = [100, 99, 98, \ldots, 2, 1]$
Dari hasil pengujian tersebut didapatkan hasil sebagai berikut :

```
==========================================
WORST CASE
Banyak data : 100
Iterative selection sort dieksekusi dalam 0.000434 detik
hasil :  [91, 92, 93, 94, 95, 96, 97, 98, 99, 100]
Recursive selection sort dieksekusi dalam 0.000436 detik
hasil :  [91, 92, 93, 94, 95, 96, 97, 98, 99, 100]
==========================================
BEST CASE
Banyak data : 100
Iterative selection sort dieksekusi dalam 0.000381 detik
hasil :  [90, 91, 92, 93, 94, 95, 96, 97, 98, 99]
Recursive selection sort dieksekusi dalam 0.000455 detik
hasil :  [90, 91, 92, 93, 94, 95, 96, 97, 98, 99]
==========================================
```

Dari pengujian worst dan best case mendapatkan kesimpulan bahwa tidak ada worst dan best case tidak ada dalam algoritma selection sort

3 Pengujian

Melakukan pengujian terhadap input size 10, 100, 500, 1000, dan 1500 acak.
Data yang diujikan sebagai berikut :
$10 = [480, 699, 1172, \ldots, 210, 1511]$
$100 = [991, 403, 298, 169, \ldots, 41, 178, 984]$
$500 = [156, 2, 1043, 308, 1415, \ldots, 891, 526, 99]$
$1000 = [397, 1507, 727, 466, \ldots, 504, 1417, 1613]$
$1500 = [719, 471, 319, 510, \ldots, 193, 593, 819]$
Dari hasil pengujian tersebut didapatkan hasil sebagai berikut :

```
========================
Banyak data : 10
Iterative selection sort dieksekusi dalam 0.000020 detik
hasil :  [210, 480, 699, 928, 944, 1172, 1509, 1511, 1540, 1627]
Recursive selection sort dieksekusi dalam 0.000019 detik
hasil :  [210, 480, 699, 928, 944, 1172, 1509, 1511, 1540, 1627]
========================
Banyak data : 100
Iterative selection sort dieksekusi dalam 0.000337 detik
hasil :  [1407, 1469, 1521, 1525, 1565, 1578, 1591, 1606, 1626, 1647]
Recursive selection sort dieksekusi dalam 0.000385 detik
hasil :  [1407, 1469, 1521, 1525, 1565, 1578, 1591, 1606, 1626, 1647]
========================
Banyak data : 500
Iterative selection sort dieksekusi dalam 0.008802 detik
hasil :  [1658, 1663, 1672, 1673, 1682, 1684, 1684, 1689, 1694, 1696]
Recursive selection sort dieksekusi dalam 0.009498 detik
hasil :  [1658, 1663, 1672, 1673, 1682, 1684, 1684, 1689, 1694, 1696]
========================
Banyak data : 1000
Iterative selection sort dieksekusi dalam 0.040314 detik
hasil :  [1683, 1683, 1688, 1690, 1690, 1691, 1691, 1695, 1695, 1698]
Recursive selection sort dieksekusi dalam 0.039260 detik
hasil :  [1683, 1683, 1688, 1690, 1690, 1691, 1691, 1695, 1695, 1698]
```

```
=========================
Banyak data : 1500
Iterative selection sort dieksekusi dalam 0.084786 detik
hasil :  [1692, 1692, 1692, 1693, 1694, 1695, 1696, 1696, 1698, 1699]
Recursive selection sort dieksekusi dalam 0.089137 detik
hasil :  [1692, 1692, 1692, 1693, 1694, 1695, 1696, 1696, 1698, 1699]
=========================
```

4 Analisis Kompleksitas

4.1 Basic operation

```
for j=i+1 to n do
        if A[j] < A[min] then
                min = j
```

Figure 3: basic operation iterative selection sort

```
temp = A[min]
A[min] = A[i]
A[i] = temp
selectionSortRecursive(A,i+1)
```

Figure 4: basic operation recursive selection sort

4.2 Perhitungan kompleksitas waktu

4.2.1 Iterative

$$
\begin{aligned}
C(n) &= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} 1 = \sum_{i=1}^{n-1} n - (i+1) + 1 = \sum_{i=1}^{n-1} n - i \\
&= n \sum_{i=1}^{n-1} 1 - \sum_{i=1}^{n-1} i = n \sum_{i=1}^{n-1} 1 - \frac{(n-1)n}{2} \\
&= n(n-1) - \frac{(n-1)n}{2} \\
&= \frac{(n-1)n}{2} \approx \frac{1}{2} n^2 \in O(n^2) \qquad (1)
\end{aligned}
$$

4.2.2 Recursive

$$M(n) = \begin{cases} 0 & n \leq 1 \\ M(n-1) + n - 1 & n > 1 \end{cases} \tag{2}$$

$$\begin{aligned} M(n) &= M(n-1) + n - 1 \\ &= (M(n-2) + n - 1) + n - 1 \\ &= M(n-2) + 2n - 2 \\ &= (M(n-3) + n - 1) + 2n - 2 \\ &= M(n-3) + 3n - 3 \\ &\vdots \\ &= M(n-i) + in - i \\ n - i &= 1, \text{ sehingga } i = n - 1 \end{aligned} \tag{3}$$

$$\begin{aligned} \text{maka, } M(n-i) + in - i &= M(n-(n-1)) + (n-1)n - (n-1) \\ &= M(n-n+1) + n^2 - n - n + 1 \\ &= M(1) + n^2 - 2n + 1 \\ &= 0 + n^2 - 2n + 1 \\ &= n^2 - 2n + 1 \in O(n^2) \end{aligned} \tag{4}$$

4.3 Notasi Asimtotik

Algoritma selection sort memiliki notasi asimtotik yang sama baik iterative maupun recursive yaitu $O(n^2)$

4.4 Class Efficiency

Algoritma selection sort memiliki class efficiency yaitu **Quadratic**

5 Analisis Tambahan

Dari hasil pengujian didapatkan data sebagai berikut :

N	Iterative (s)	Recursive (s)
10	0.000020	0.000019
100	0.000337	0.000385
500	0.008802	0.009498
1000	0.040314	0.039260
1500	0.084786	0.089137

Table 1: hasil pengujian execution time

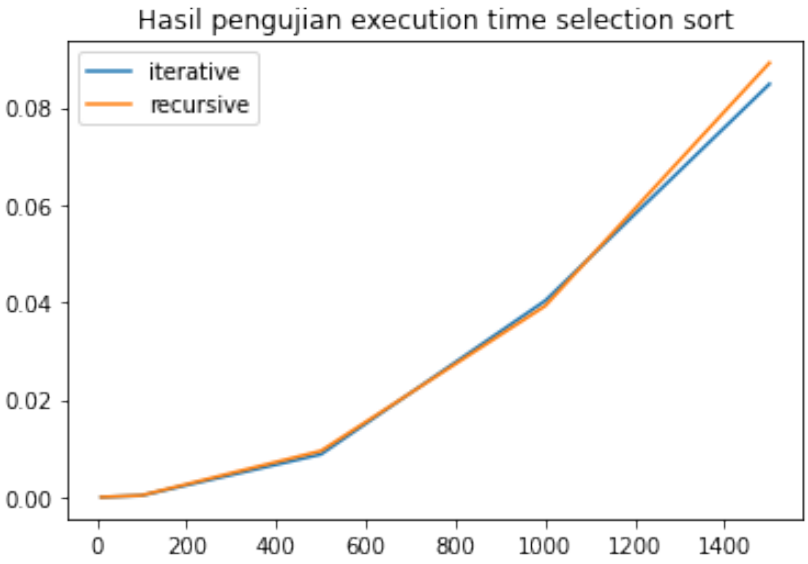

Figure 5: line plot hasil pengujian execution time

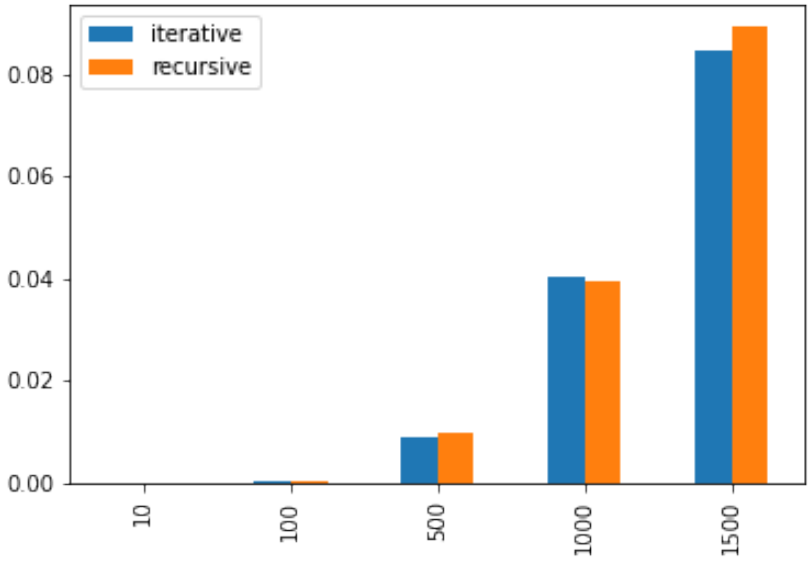

Figure 6: bar plot hasil pengujian execution time

6 Kesimpulan

Dapat disimpulkan bahwa :

1. Tidak memiliki best case dan worst case
2. Kompleksitas algoritma selection sort adalah $O(n^2)$

7 Sumber dan Referensi

1. Algoritma Selection Sort
2. Repository github Tubes-AKA
3. Google Colaboratory Tubes-AKA