Analyses of various algorithms

Term paper for INF221 2021

Mats Hoem Olsen, Roy Erling Granheim, Sindre Stokke

Norwegian University of Life Sciences

DataSci

Norwegian university of life science

2021 December 3

Contents

Abstract

algorithm goes brrrr.

1 Introduction

Hi, we are students. How are you?(see Cormen et al., *Introduction to Algorithms*)

2 Theory

Provide a brief description of the algorithms you will be investigating, including pseudocode for the algorithms. Describe in particular the expected runtime of algorithms in terms of problem size. Use a separate subsection for each algorithm.

2.1 Algo 1: Bubble sort

Algorithm 1 Insertion sort algorithm from Cormen et al., *Introduction to Algorithms*, Ch. 2.1.

```
BUBBLESORT(A)
for i = 1 to A.length − 1
    for j = A.length downto i + 1
        if A[j] < A[j − 1]
            exchange A[j] with A[j − 1]
```

Pseudocode for the first algorithm is shown in Listing 1. Best case runtime for this algorithm is

$$T(n) = \Theta(n)\ . \tag{1}$$

It is achieved for correctly sorted input data.

2.2 Algo 2: Insertion sort

Algorithm 2 Insertion sort algorithm from Cormen et al., *Introduction to Algorithms*, Ch. 2.1.

```
INSERTION-SORT(A)
for j = 2 to A.length
    key = A[j]
    i = j − 1
    while i > 0 and A[i] > key
        A[i + 1] = A[i]
        i = i − 1
    A[i + 1] = key
```

Pseudocode for the second algorithm is shown in Listing 2. Best case runtime for this algorithm is

$$T(n) = \Theta(n)\ . \tag{2}$$

It is achieved for correctly sorted input data.

2.3 Algo 3: Quicksort

Algorithm 3 Quicksort algorithm from Cormen et al., *Introduction to Algorithms*, Ch. 2.1.

```
QUICKSORT(A, p, r)
if p < r
    q = PARTITION(A, p, r)
    QUICKSORT(A, p, q − 1)
    QUICKSORT(A, q + 1, r)
```

Algorithm 4 Partition from Cormen et al., *Introduction to Algorithms*, Ch. 2.1.

```
PARTITION(A, p, r)
x = A[r]
i = p − 1
for j = p to r − 1
    if A[j] ≤ x
        i = i + 1
        exchange A[i] with A[j]
exchange A[i + 1] with A[r]
return i + 1
```

Pseudocode for the second algorithm is shown in Listing 4. Best case runtime for this algorithm is

$$T(n) = \Theta(n) \ . \tag{3}$$

It is achieved for correctly sorted input data.

2.4 Algo 4: Quicksort Median of Three

Algorithm 5 Quicksort median of three algorithm from Cormen et al., *Introduction to Algorithms*, Ch. 2.1.

```
INSERTION-SORT(A)
for j = 2 to A.length
    key = A[j]
    i = j − 1
    while i > 0 and A[i] > key
        A[i + 1] = A[i]
        i = i − 1
    A[i + 1] = key
```

Pseudocode for the second algorithm is shown in Listing 5. Best case runtime for this algorithm is

$$T(n) = \Theta(n) \ . \tag{4}$$

It is achieved for correctly sorted input data.

2.5 Algo 5: Quicksort Insertion sort hybrid

Algorithm 6 Insertion sort algorithm from Cormen et al., *Introduction to Algorithms*, Ch. 2.1.

```
INSERTION-SORT(A)
for j = 2 to A.length
    key = A[j]
    i = j − 1
    while i > 0 and A[i] > key
        A[i + 1] = A[i]
        i = i − 1
    A[i + 1] = key
```

Pseudocode for the second algorithm is shown in Listing 6. Best case runtime for this algorithm is

$$T(n) = \Theta(n) \ . \tag{5}$$

It is achieved for correctly sorted input data.

2.6 Algo 6: Mergesort

why are you like this

Algorithm 7 Mergesort algorithm from Cormen et al., *Introduction to Algorithms*, Ch. 2.1.

```
MERGE-SORT(A, p, r)
if p < r
    q = ⌊(p + r)/2⌋
    MERGE-SORT(A, p, q)
    MERGE-SORT(A, q + 1, r)
    MERGE(A, p, q, r)
```

Algorithm 8 Merge from Cormen et al., *Introduction to Algorithms*, Ch. 2.1.

```
MERGE(A, p, q, r)
n1 = q − p + 1
n2 = r − q
let L[1..n1 + 1] and R[1..n2 + 1] be new arrays
for i = 1 to n1
    L[i] = A[p + i − 1]
for j = 1 to n2
    R[j] = A[q + j]
L[n1 + 1] = ∞
R[n2 + 1] = ∞
i = 1
j = 1
for k = p to r
    if L[i] ≤ R[j]
        A[k] = L[i]
        i = i + 1
    else A[k] = R[j]
        j = j + 1
```

Pseudocode for the second algorithm is shown in Listing 8. Best case runtime for this algorithm is

$$T(n) = \Theta(n)\ . \tag{6}$$

It is achieved for correctly sorted input data.

2.7 Algo 7: Mergesort Insertion sort hybrid

Algorithm 9 Insertion sort algorithm from Cormen et al., *Introduction to Algorithms*, Ch. 2.1.

```
INSERTION-SORT(A)
for j = 2 to A.length
    key = A[j]
    i = j − 1
    while i > 0 and A[i] > key
        A[i + 1] = A[i]
        i = i − 1
    A[i + 1] = key
```

Pseudocode for the second algorithm is shown in Listing 9. Best case runtime for this algorithm is

$$T(n) = \Theta(n)\ . \tag{7}$$

It is achieved for correctly sorted input data.

2.8 Algo 8: Python sort

Best case runtime for this algorithm is

$$T(n) = \Theta(n)\ . \tag{8}$$

It is achieved for correctly sorted input data.

2.9 Algo 9: Numpy sort

Best case runtime for this algorithm is

$$T(n) = \Theta(n)\ . \tag{9}$$

It is achieved for correctly sorted input data.

3 Method used

In this section we will show our workflow and automation of tests.(**brady_haran_death_2020**)

4 Benchmarks

In this section we will analyse the performance of the different algorithms on exponentially growing test data.

5 Discussion

We will in this section discuss our findings, and thoughts[1]

6 Acknowledgements

We acknowledge that we may, or may not be made out of bread.

References

Cormen, Thomas H. et al. *Introduction to Algorithms*. 3rd. Computer science. USA: McGraw-Hill, 2009. 1292 pp. ISBN: 978-0-262-03384-8. URL: `https://books.google.no/books?id=aefUBQAAQBAJ`.

[1] Although these thoughts are subjective, we will present them as facts. For our thoughts feel real to us.