Module 3 Critical Thinking: Minimum Number

Donghwan Kim

Colorado State University Global

CSC 506: Design and Analysis of Algorithms

Dr. Lori Farr

May 12, 2022

Module 3 Critical Thinking: Minimum Number

This assignment discusses algorithms to find the minimum value of a sequence of numbers in a list. The `min` function is one of operations widely used in many applications. It designs several algorithms including the one which compares each number to every other number on the list. Iterative O(n^2) and O(n) functions are discussed along with the algorithms with recursive methods. It also discusses data structures for a sequence of numbers.

Understanding the problem

The `min` function, one of the most used operations in mathematics and statistics, returns the smallest number in the data. This problem to find the minimum number can start with comparing any two numbers. Given any two numbers a and b, the minimum function, `min`(a, b), is equal to b if $b < a$ and a otherwise. It can be designed like Algorithm (1). The minimum is also expressed as $\min(a, b) = 1/2 * (a + b - |a - b|)$ where $|a - b|$ is the absolute value.

```
Algorithm (1)
1. Function MinInTwoNumbers(a,b)
2.     //input: two numbers
3.     //output: a number
4.     if a > b: return b
5.     else: return
```

As shown in Algorithm (2), the minimum value in a list can be found using the two iterations with indices i and j each of which goes through the elements of the list. It compares each number to every other number. To do that, the algorithm has two nested loops that each iterate O(n) times, so the runtime is O(n^2) runtime (Lysecky and Vahid 2019: 2.6 Algorithm analysis). Algorithm (3) below is an iterative O(n) min function.

```
Algorithm (2)
1. Function MinInList(A)
```

```
2.     //input: a list, output: a number
3.     min_value <- A[1]
4.     for i in 1 to n DO:
5.         for j in 1 to n DO:
6.             if A[j] != A[i] and MinInTwoNumbers(A[i], A[j]) < min_value:
7.                 min_value = MinInTwoNumbers(A[i], A[j])
8.     return min_value
```

Algorithm (3)

```
1. def MinInList(A):
2.    min_value = A[1]
3.    for i == 2 to n DO:
4.         min_value = MinInTwoNumbers(min_value, A[i])
5.    return min_value
```

Implementation of the algorithms requires a list for storing a sequence of numbers. Python provides a data type called list which can store a collection of elements including numbers Van Rossum and Drake Jr (2009). As shown in the algorithms, the elements of a Python list can be accessed using the index. Fortran 90/95 has data type for a sequence of elements which is called array (Wagener 1998).

Other algorithms for the min function

There are many other ways to solve the problem. Algorithm (4) and (5) use the recursive method to find the minimum. Another approach is to a sort algorithm which orders the numbers in a ascending or descending order. In fact, the selection sort algorithm using the method of finding the smallest number in the design of sort.

Algorithm (4)

```
1. Function MinInList(A):
2.     if len(A) == 1:
```

```
3.          return A[1]
4.      else:
5.          if A[1] < MinInList(A[2:]):
6.              return A[1]
7.          else:
8.              return MinInList(A[2:])
```

Algorithm (5)

```
1. Function MinInList(A, n):
2.      if n == 1:
3.          return A[1]
4.      else:
5.          return MinInTwoNumbers(A[1], MinInList(A[2;], n-1))
```

Conclusions

The `min` function is widely used in many applications and the minimum-finding problem can be extended to other related optimization problems. This assignment discusses algorithms to find the minimum numbers in a list. It designs iterative O(n^2) and O(n) algorithms of the function . It also shows algorithms utilizing recursive algorithm. Data type to store a sequence of numbers are discussed. As an extension, a special case which can be thought of is to find the minimum when each of the elements of a list is gradually decreasing and then increasing or when a list has local and global minimums.

References

Lysecky, R. and Vahid, F. (2019). Design and analysis of algorithms. In Lysecky, R. and Vahid, F., editors, *Data structures essential: Pseudocode with python examples*. Zybooks.

Van Rossum, G. and Drake Jr, F. L. (2009). *Python tutorial*. Scotts Valley, CA: CreateSpace.

Wagener, J. (1998). *Fortran 90 concise reference*. Absoft Corporation.