Exercise of Getting Started

Grabur

Feb 2025

1 Chapter 1: Foundations

1.1 The Role of Algorithms in Computing

I didn't make the exercise of this section because I didn't find them useful.

1.2 Getting Started

Exercise 1.2-1):
It could be an application like booking. When you search a hotel close to the airport, it gets involved algorithms as searching the hotels close to that airport and it should be searched in a short time period.

Exercise 1.2-2):

$$8n^2 < 64n \cdot \log_2 n \quad \rightarrow \quad n < 8 \cdot \log_2 n$$

Try values until this inequality is false. To $n \lessapprox 43$, insertion sort runs faster than merge sort.

Exercise 1.2-3):

$$100n^2 < 2^n$$

Trying values, for $n \lessapprox 15$, 2^n runs faster than $100n^2$.

Exercise 2.1-1):
Note: Resolved using the logic of C, C++, Java, etc. while iterating over an array on a for loop. Also the number that appears in green, is the number being checked. The number or numbers that appears in red are the numbers being moved.

i	Array
1)	[31, 41, 59, 26, 41, 58]
2)	[31, 41, 59, 26, 41, 58]
3)	[26, 31, 41, 59, 41, 58]
4)	[26, 31, 41, 41, 59, 58]
5)	[26, 31, 41, 41, 58, 59]

Exercise 2.1-2):

Initialization: The loop start getting the first number in the array. In spite of that, it has initialized to 0 the variable sum where the total sum will be stored. Due to that, the invariant holds the first number that will be added to sum.

Maintenance: On each iteration, the loop will hold only the index of the number that will be added, after add it, i will be incremented by 1, holding the next number (i + 1).

Termination: The loop will terminate when the 'n' elements of the array are added. In conclusion, sum it's equivalent of say that $sum = \sum_{i=1}^{n} A[i]$