UWI

Design and Analysis of Algorithms

Tutorial 1

Time Efficiency

In this course, emphasis is placed on analysing algorithms by their time efficiency. Knowing an algorithm's time efficiency can encourage us to make them more efficient by reducing the number of iterations to perform the algorithm's task.

1. What is time efficiency?
2. What is a basic operation?
3. Given the following formula $T(n) \approx c_{op}C(n)$
   (a) What does n mean?
   (b) What does c_{op} mean?
   (c) What does $T(n)$ mean?
   (d) What does $C(n)$ mean?
4. Given that $C_b(n)$, $C_w(n)$ and $C_a(n)$ stand for the best, worst and average case time complexities.
   (a) What leads to the best case?
   (b) What leads to the worst case?
   (c) What is meant by the average case?
5. Asymptotic orders of growth serve as ways of comparing the efficiency groups of algorithms.
   (a) What are the 3 asymptotic notations and what do they mean?
   (b) What are the 8 basic asymptotic efficiency classes? Indicate the most and least efficient ones.
6. Analyse the following algorithms and identify their basic operations. Discuss their best and worst case time efficiencies.
   (a) Algorithm 1: Sequential Search

```
for i ← 0 to n–1 do
    if A[i] = K then
        return i
    end if
end for
return –1
```

   (b) Algorithm 2: Max Element

```
maxval ← A[0]
for i ← 1 to n–1 do
    if A[i] > maxval then
        maxval ← A[i]
    end if
end for
return maxval
```

Mathematical Proofs

In this course, similar emphasis is also placed on mathematical proofs. This is because they serve as a foundation for some subsequent algorithmic proofs.

1. Prove that $\lfloor log_2(n) \rfloor + 1 = \lceil log_2(n+1) \rceil$, where n is a positive integer.