Graphs and Complexity for IoT

Exercice 1:

Sequential/linear vs. dichotomic/binary search

Exercice 2:

Bubble vs. Merge sort

Exercice 3:

Consider two algorithms A1 and A2 with their respective execution times

T1(n) = 9n^2 and T2(n) = 100n + 96 respectively.

1. Determine the asymptotic complexity of the two algorithms.
2. Which algorithm has the best asymptotic complexity?
3. Show that your solutions are correct by specifying c and n0 values so that the Upper Bound notation is satisfied (see the course material).
4. For which data length n, A1 or A2 is the most efficient?
5. What is the asymptotic complexity of the following algorithm? What rules did you apply?

```
Begin
      call A1 ;
      call A2 ;
End
```

Exercice 4 : Stacking problem

A cable manufacturer has only a single reel/coil of K metres. He has received N orders, each consisting of a certain length of cable, which cannot be fragmented. He wants to maximise the total length of the cable he can serve with the single remaining reel/coil. Here's an application example:

- N =10
- Orders = {2463, 2597, 315, 1089, 759, 599, 1283, 1683, 1625, 2292, 211}
- K =6790

A solution: 6790 = 2597 + 315 + 759 + 1283 + 1625 + 211

-- Measure the execution time for sets of size 10, 20, 30, ...
-- What is the complexity of your algorithm?

Exercice 5: Matrix addition and multiplication

Consider two square matrices A and B of size n :

1. Write an algorithm that adds the two matrix A and B.
2. Write an algorithm to multiply the two matrix A and B.
3. Determine the "worst case" time function T(n) for matrices of size n.
4. Determine the asymptotic complexity for the two algorithms.

Exercice 6 :
Consider the following algorithms with execution time T(n) for data size n.
Determine their respective asymptotic complexity.

- **Algorithm A1**: T(n) = 3n + 2
- **Algorithm A2**: T(n) = 6
- **Algorithm A3**: T(n) = 4n^2 + n + 2

- **Algorithm A4** :

```
execute A1;
execute A2;
execute A3;
```

- **Algorithm A5** :

```
For i from 1 to n do
        execute A3;
Endfor
execute A1;
```

- **Algorithm A6** :

```
For i from 1 to 5 do
        execute A1;
Endfor
```

Exercice 7 :
Consider the following algorithms

Algorithm 1	Algorithm 2

1. x = n ;
2. while (x > 0) {
3. y = n ;
4. while (y > 0) {
5. y = y – 1 ;
6. }
7. x = x / 2 ;
8. }
	1. x = n ;
	2. while (x > 0) {
	3. y = x ;
	4. while (y > 0) {
	5. y = y – 1 ;
	6. }
	7. x = x / 2 ;
	8. }

Calculate the asymptotic complexity of the two algorithms? Justify?

Exercice 8: TSP problem

Objectives :
-- become familiar with graph algorithms,
-- evaluate the impact of the choice of solution and data structures on execution time,
-- evaluate the time complexity of the proposed algorithms,
-- become familiar with the famous travelling salesman problem.

The travelling salesman's problem consists of finding the order of visits of a set of cities that minimises the total distance travelled by the traveller. It can be applied to a wide range of problems, including logistics, transport and scheduling. In this exercice, we will study two variants of the problem. For example, we can look for the shortest route connecting the main cities in France. The problem is defined by an undirected, valuated graph G = (V, E) with n vertices. Each vertex represents a city, and each edge (i, j) ∈ E, a route between cities i and j; the value associated with (i, j) is the distance of the route between i and j.

Part 1: The aim of the first two questions is to check that the graph is connected and to construct a matrix of distances between all the cities.

Q1: Define a function that can generate a random graph.

Q2: Define a function that checks whether the graph of cities is connected.

Q3: Now assume that the graph is connected. We want to calculate all the lengths of the shortest paths between all the cities. These distances will be stored in a distance[ ][ ] matrix.

Note: you can use Dijkstra's algorithm

Part 2: We are now interested in the complete graph (a graph is complete if each pair of vertices is connected by an edge) given by distance[ ][ ]. In this complete graph, we are looking for a tour, i.e. a cycle that passes through each of the cities once and only once before returning to its starting point. The length of a tour is given by the sum of the distances travelled through the cities, including the return to the starting point.

Two solutions:

Q4: Simple approach (nearest neighbor first): the simplest constructive solution is to start from one town and then go from town to town, visiting the nearest unvisited town each time.

Q5: Exact approach that returns an optimal solution.
Q6: Compare the execution times and the obtained length tour of the two variants for a number of cities ranging from 10 to 20.

Exercice 9: Some nice and useful graph problems for networks.

- MIS
- MDS
- Vetex and Edge Cover
- Graph coloring