Le 6 – Graph Algorithms and Algorithmic Paradigms

1. Draw a connected, undirected, weighted graph with 8 vertices and 16 edges, each with unique edge weights.
   - Illustrate the execution of Kruskal’s algorithm on this graph.
   - Identify one vertex as a “start” vertex and illustrate the execution of Dijkstra’s algorithm on this graph.

2. John is an exchange student in city L. He wants to go for an excursion but he can only pay at most 200 in local currency for the ticket. The following graph shows the excursion ticket prices between some destinations that may be interesting for John. The cost of excursion along an arbitrary path in the graph is the sum of the edge costs.

   (a) Give the idea of an algorithm that would find all destinations that John can afford.
   (b) Illustrate the operation of the algorithm on the example graph; show the cost for each affordable destination.

3. Members of the yacht club have their summer houses at six neighbouring islands in St. Anna archipelago. The distances between the islands are so small that they can be connected by bridges. The cost of a bridge depends on the distance. The distances in meters are given in the table:

   |   | 12 | 13 | 14 | 15 | 16 |
---|---|---|---|---|---|
11 | 52 | 28 | 26 | 25 | 43 |
12 | 75 | 26 | 50 | 81 |
13 | 21 | 10 | 5  |
14 | 15 | 95 |
15 | 12 |

Decide which bridges are to be built to connect all islands at minimum cost. Justify your decision and show step by step how it was computed.

4. Outline an algorithm that for a given vertex \( v \) of a weighted DAG \( G \) with \( n \) vertices and \( m \) edges computes the length of the shortest path from \( v \) to every vertex of \( G \) in time \( O(m + n) \).

5. The timetable of daily flights shows the direct non-stop flights between some of the airports A, B, C and D. e.g.
   - A to B: 8.30, 12.30, 18.30
   - B to A: 6.30, 10.30, 15.15
   - Duration of each flight 1 hour 15 min.
• A to C: 10.15, 17.30
  C to A: 8.30, 15.15
  Duration of each flight 1h 45 min.
• B to C: 11.15, 13.25, 18.00
  C to B: 9.30, 12.50, 16.15
  Duration of each flight 45 min.
• B to D: 11.15, 14.25, 18.00
  D to B: 9.30, 16.15, 20.30
  Duration of each flight 1 hour 10 min.
• C to D: 11.15, 18.00
  D to C: 9.30, 16.15
  Duration of each flight 50 min.

A connecting flight at any of the airports can be booked not earlier than 45 min after the arrival.

For given airports X and Y we want to find a flight connection that leaves X at earliest at time t and arrives Y as early as possible.

(a) Formulate the problem as a graph problem. Which of the known graph algorithms can be modified to solve it? Justify your answer and discuss the modifications needed.

(b) Illustrate the steps of the proposed algorithm for the example timetable for a trip from A to D with departure at earliest at 10.00.

(c) Discuss the complexity of your algorithm.

6. An n-degree polynomial is an expression of the form
\[ a_0 + a_1 x + \ldots + a_n x^n \]
where \( a_0, \ldots, a_n \) are constants \( a_n \neq 0 \) and \( x \) is a variable ranging over real numbers.

(a) Analyze the time complexity of computing the value of this expression following its form, under the assumption that \( x^k \) is not a primitive operation but has to be computed by multiplication.

(b) Outline another algorithm that under the same assumption computes the value of a polynomial of degree \( n \) in time \( \Theta(n) \). Can you link this solution to one of the algorithm paradigms discussed in the course?

7. Explain the principles of greedy method and justify why Huffman’s algorithm for building optimal encodings is a greedy algorithm. What are other examples of greedy algorithms discussed in this course?

8. Draw the frequency array and Huffman tree for the following string

   'dogs do not spot hot pots or cats'

   Show the encoding of the string 'post stop'.

9. Explain what is the principle underlying the longest common subsequence algorithm. Construct the longest common subsequence array for the strings 'XZACKDFWG' and 'ABCDEEFG'. What is the longest common subsequence of these strings?

10. Consider the usual definition of Fibonacci numbers:
    \[ F(0) = 0, \quad F(1) = 1, \quad F(n) = F(n - 1) + F(n - 2). \]
    Analyze the time complexity of a recursive procedure using this definition for computing Fibonacci numbers. Design an algorithm for computing Fibonacci numbers based on the dynamic programming principle and analyze its time complexity.
11. Consider the following pseudocode describing a procedure foo referring to an auxiliary
procedure bar.

\[
\text{procedure foo(array a[l..r])}
\]
\[
\text{if } l < r \text{ then}
\]
\[
m \leftarrow (l + r)/2
\]
\[
\text{foo(a[l..m])}
\]
\[
\text{foo(a[m+1..r])}
\]
\[
\text{bar(a[l..m], a[m+1..r])}
\]

(a) Which algorithm paradigm is represented by this procedure?

(b) Which of the algorithms discussed in the course has the same structure as this
procedure.

12. Consider the following problem. The input data consists of \( n \geq 1 \) rows of integers from
the interval \([-\text{maxint}, \text{maxint}]\). The \( i \)-th row has \( i \) elements, for \( i = 1, \ldots, n \). Thus, an
input data can be seen as a triangle, as illustrated below for \( n = 4 \).

\[
\begin{array}{cccc}
\text{x1,1} & \text{x2,1} & \text{x2,2} \\
\text{x3,1} & \text{x3,2} & \text{x3,3} \\
\text{x4,1} & \text{x4,2} & \text{x4,3} & \text{x4,4} \\
\end{array}
\]

A path in such a triangle is a sequence of numbers obtained by starting at the top and
choosing at every step one of the two closest elements in the next row until the last
row is reached. More precisely, it is a sequence of \( n \) numbers that starts with \( x_{1,1} \) and
such that each of its elements \( x_{i,j} \) is followed by the element \( x_{i+1,j} \) or by the element
\( x_{i+1,j+1} \), for \( i = 1, \ldots, n - 1 \). For the example above, the following are the paths that
end with the number \( x_{4,2} \):

\[
\begin{array}{c}
x_{1,1}, x_{2,1}, x_{3,1}, x_{4,2} \\
x_{1,1}, x_{2,1}, x_{3,2}, x_{4,2} \\
x_{1,1}, x_{2,2}, x_{3,2}, x_{4,2} \\
\end{array}
\]

A cost of a path is the sum of its elements. The problem is to find the cost of the most
expensive path in a triangle of size \( n \).

(a) Construct a recursive algorithm that solves the problem by starting at the top
of the triangle. Analyze the time complexity of this algorithm (upper bound and
lower bound).

(b) Construct another algorithm that solves the problem in \( \Theta(n) \).

(c) Which are the underlying algorithm paradigms of these algorithms?