University of Padova
Master's degree in Computer Science

Advanced Algorithms

Spring 2023

July 6, 2023 – 14:30–16:30

First Part: Theory Questions

Question 1 (4 points) Give a list of graph problems that can be solved in time $O(m+n)$ applying DFS or BFS. Your list should include at least four problems to receive full points.

Question 2 (4 points) Consider the following weighted graph, represented by an adjacency matrix where each numerical value represents the weight of the corresponding edge, and where the symbol '–' indicates the absence of the edge between the corresponding vertices.

	a	b	c	d	e	f
a	-	3	-	5	7	9
b		-	8	6	-	-
c			-	4	-	-
d				-	2	-
e					-	1
f						-

(a) Draw the graph.

(b) List the edges of the minimum spanning tree in the order they are selected by Kruskal's algorithm.

(c) List the edges of the minimum spanning tree in the order they are selected by Prim's algorithm starting at vertex a.

Question 3 (4 points) For each of the following problems, say whether it is NP-hard or not and, if not, specify the complexity of the best algorithm seen in class.

(a) Maximum flow

(b) All-pairs shortest paths

(c) Vertex cover

(d) Single-source shortest paths

Second Part: Problem Solving

Exercise 1 (10 points) In the *bin packing* problem we are given a set of n items, each with weight between 0 and 1, and we are asked to load the items into as few bins as possible, such that the total weight in each bin is at most 1. This problem is NP-hard; this exercise asks you to analyze the following simple approximation algorithm, where the input is an array $W[1 \ldots n]$ of weights and the output is the number of bins used.

```
ApproxBinPacking(W[1...n])
   b = 1                \\ b = current bin
   Total[1] = W[1]      \\ Total[b] = total weight of bin b
   for i = 2 to n do
      if Total[b] + W[i] <= 1 then      \\ item i fits into the current bin
         Total[b] = Total[b] + W[i]     \\ item i is placed inside it
      else
         b = b + 1            \\ a new bin (initially empty) is opened
         Total[b] = W[i]      \\ item i is placed inside this new bin
   return b
```

(a) Prove that this algorithm is a 2-approximation algorithm. (Hint: consider pairs of bins...)

(b) Prove a lower bound to the approximation factor of this algorithm. To receive full points the lower bound should be 2.

Exercise 2 (10 points) Suppose you have a randomized algorithm for a minimization problem A that returns the correct output with probability at least $1/n$, where n is the input size. Show how to obtain an algorithm for A that returns the correct output with high probability. (Hint: for the analysis use this inequality: $(1 + x/y)^y \leq e^x$ for $y \geq 1$, $y \geq x$.)