University of Padova
Master's degree in Computer Science

Advanced Algorithms

Spring 2023

June 22, 2023 – 14:30–16:30

First Part: Theory Questions

Question 1 (4 points) Consider the following directed, 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.

	s	a	b	c	d
s	-	2	4	-	-
a	-	-	-1	2	-
b	-	-	-	-	4
c	-	-	-	-	2
d	-	-	-	-	-

(a) Draw the graph.

(b) Run the Bellman-Ford algorithm on this graph, using vertex s as the source. You are to return the trace of the execution, i.e. a table with rows indexed by vertices and columns indexed by iteration indexes (starting from 0) where each entry contains the estimated distance between s and that vertex at that iteration.

Question 2 (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 independent set

(b) All-pairs shortest paths

(c) Traveling salesperson problem

(d) Graph connectivity

Question 3 (4 points) Define the set cover problem and briefly describe the $O(\log n)$-approximation algorithm seen in class.

Second Part: Problem Solving

Exercise 1 (9 points) Consider Dijkstra's algorithm seen in class, which returns the lengths of the shortest paths from a source vertex to all other vertices in directed graphs with nonnegative weights:

(a) Explain how to modify Dijkstra's algorithm to return the shortest paths themselves (and not just their lengths).

(b) Consider the following algorithm for finding shortest paths in a directed graph where edges may have negative weights: add the same large constant to each edge weight so that all the weights become nonnegative, then run Dijkstra's algorithm and return the shortest paths. Is this a valid method? Either prove that it works (i.e., the returned shortest paths are shortest paths in the original graph), or give a counterexample.

(c) Now let's switch to minimum spanning trees, and do the same: add the same large constant to each edge weight and then run Prim's algorithm. Either prove that the returned solution is a minimum spanning tree of the original graph, or give a counterexample.

Exercise 2 (10 points) Suppose you throw n balls into $\frac{n}{6 \ln n}$ bins[1] independently and uniformly at random. Applying the following Chernoff bound show that, with high probability, the bin with maximum load (load = number of balls in the bin) contains at most $12 \ln n$ balls. (Hint: focus first on one arbitrary bin and bound the probability of that bin's load exceeding $12 \ln n$ ...)

Theorem 1. *Let $X_1, X_2, \ldots, X_n$ be independent indicator random variables such that $E[X_i] = p_i, 0 < p_i < 1$. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = E[X]$. Then, for $0 < \delta \leq 1$,*

$$\Pr(X > (1+\delta)\mu) \leq e^{-\mu\delta^2/3}.$$

[1] Recall that $\ln n = \log_e n$.