1 Advanced Algorithms (tms41)

(*a*) Suppose you have a randomised approximation algorithm for a maximisation problem such that, for any $\epsilon > 0$ and any problem instance of size n, the algorithm returns a solution with cost C such that

$$\mathbf{Pr}[C \geq (1 - 1/\epsilon) \cdot C^*] \geq 1/n \cdot \exp(-1/\epsilon),$$

where C^* is the cost of the optimal solution. Can you use your algorithm to obtain a PTAS or FTPAS? Justify your answer. [6 marks]

(*b*) We consider the following optimisation problem. Given an undirected graph $G = (V, E)$ with non-negative edge weights $w : E \to \mathbb{R}^+$, we are looking for an assignment of vertex weights $x : V \to \mathbb{R}$ such that: (*i*) for every edge $\{u, v\} \in E$, $x(u) + x(v) \geq w(\{u, v\})$, (*ii*) $\sum_{v \in V} x(v)$ is as small as possible.

(*i*) Design a 2-approximation algorithm for this problem. Also analyse the running time and prove the upper bound on the approximation ratio.
Note: For full marks, your algorithm should run in at most $O(E^2)$ time.

Hint: One way to solve this question is to follow the approach used by the greedy approximation algorithm for the VERTEX-COVER problem. [8 marks]

(*ii*) Can this problem be solved exactly in polynomial-time? Either describe the algorithm (including a justification of its correctness and why it is polynomial time) or prove that the problem is hard via a suitable reduction. [6 marks]