CS6100 - Midsem - Q1

a) Given that an algorithm A solves the mentioned TSP with edge weights $\in \{1, 2\}$ in polynomial time. Now we formulate an algorithm for Hamiltonian cycle using A -

Algo: Given a graph G to find HC now construct a complete graph G' with same vertices. Now assign weight to be 1 if it is present in the original graph and 2 if not.

$$w(e) = \begin{cases} 1 & \text{if } e \in E(G) \\ 2 & \text{o.w} \end{cases}$$

Now apply algorithm A on G' to obtain TSP. If the cost of output is n then G has HC otherwise not.

proof: lets suppose G has HC then this cycle in G' has cost n that is minimum (as replacement of any edge just increases the cost). If G doesn't have a HC then there must be atleast 1 edge in TSP(G') that doesn't belong to G and this makes the cost of TSP $\geq (n-1)(1) + 1(2) = n+1$ ie $(\neq n)$

b)

Lemma-1: TSP can be formulated as linear binary optimization problem

proof: A linear binary optimization problem is a minimization/maximization of linear objective function ($c^T x = c_1 x_1 + c_2 x_2 \cdots c_n x_n$) over a arbitrary set $S \subseteq \{0,1\}^n$.

Consider TSP over a graph of n edges with weights $e_1, e_2 \cdots e_n$. Now consider a variable $x_i = 1$ if edge i is present in our path otherwise 0. Also consider set $T: \{(x_0, x_1, x_2 \cdots x_n)\}$ which consists of all hamiltonian cycles. Now our TSP boils down to minimizing $e^T x$ or $e_1 x_1 + e_2 x_2 \cdots e_n x_n$ where $x \in T$, a linear binary optimization problem.

Lemma-2: Given version of TSP is strongly n-p hard.

proof: we know from above formulation that edge weights are coefficients. For the given version - $|c_i| \in \{1,2\}$ - which means that coefficients are polynomially bounded ($\in [0, n^c]$). From part-a of this question (reduction to HC) we can say that this version of TSP is NP-hard. So by definition If a problem is np hard even when coefficients being polynomially

bounded then it is strongly NP-hard.
In literature this version of TSP is used to prove that general TSP is strongly NP-Hard.

Theorem: let Π be binary opt. problem which is strongly NP-Hard then there doesnot exist an algo for Π whose expected run time is bounded by $pol(N, \phi)$ for ϕ perturbed instances with coeff. in $[-1,1]$ unless $NP \subseteq ZPP$ (proved in class & notes)

From lemma 1 and 2 we can see that our version of TSP fits the problem Π in the theorem hence we cannot have a polynomial time algorithm of ϕ perturbed instances unless $NP = ZPP$