1.

(a) **Optimal substructure**

For any i, $0 \le i < n$, consider the first part of an optimal solution: the travel up to city i. Let j be the number of times the train is used up to city i. Then, under the restriction that the train is used j times and the plane $i - j$ times, the first part of the optimal solution gives an optimal (minimal cost) way to travel from train station 0 to

- train station i, if the optimal solution uses the train from city i to $i + 1$.
- airport i, if the optimal solution uses the plane from city i to $i + 1$.

(b) **Subproblem definition**

Define $T_{i,j}$ $(0 \le j \le i \le n)$ to be the minimum cost of travelling from train station 0 to train station i using the train j times and the plane $i - j$ times: similarly, let $A_{i,j}$ be the minimum cost of travelling to airport i.

(c) **Recurrence relation**

For $0 \le j \le i \le n$ we have

$$T_{i,j} := \begin{cases} 0 & \text{if } i = 0 \\ A_{i-1,0} + p_{i-1} + b_i & \text{if } i > 1 \text{ and } j = 0 \\ T_{i-1,j-1} + t_{i-1} & \text{if } i > 1 \text{ and } j = i \\ \min(A_{i-1,j} + p_{i-1} + b_i,\ T_{i-1,j-1} + t_{i-1}) & \text{if } i > 1 \text{ and } 0 < j < i \end{cases}$$

and

$$A_{i,j} := \begin{cases} b_0 & \text{if } i = 0 \\ A_{i-1,0} + p_{i-1} & \text{if } i > 1 \text{ and } j = 0 \\ T_{i-1,j-1} + t_{i-1} + b_i & \text{if } i > 1 \text{ and } j = i \\ \min(T_{i-1,j-1} + t_{i-1} + b_i,\ A_{i-1,j} + p_{i-1}) & \text{if } i > 1 \text{ and } 0 < j < i \end{cases}$$

(d) **Compute optimal solutions**

For $i > 0$, the recurrence expresses $T_{i,*}$ in terms of $T_{i-1,*}$ and $A_{i-1,*}$. Thus, we can initialize $T_{0,0}$ and $A_{0,0}$ and then compute $T_{i,*}$ and $A_{i,*}$ for $i = 1, 2, \ldots, n$ in succession. The solution is given by $A_{n,n/2}$. Note that for $i > n/2$ if suffices to compute $T_{i,j}$ and $A_{i,j}$ for $j \le n/2$ instead of $j \le i$, but this optimization will not improve the asymptotic running time of the algorithm.

2.

We begin by noting that SUBSET-GCD-DEC $\in$ NP. The certificate is a set C of positive indices. Given an instance $[\{a_1, \ldots, a_n\}, k]$ of SUBSET-GCD-DEC, the certificate verification algorithm checks that $C \subseteq \{1, \ldots, n\}$ with $|C| = k$, say $C = \{i_1, \ldots, i_k\}$, and that $\gcd(a_1, \ldots, a_n) = \gcd(a_{i_1}, \ldots, a_{i_k})$. The size required to encode the problem instance is $\Omega(n + \log a_1 + \cdots + \log a_n)$ bits since the encoding of a_i uses $\Theta(\log a_i)$ bits and there are n distinct a_*, each one requiring at least one bit. The certificate has size bounded by $O(n \log n)$ bits, which is polynomial in the input size, and the verification algorithm also runs in polynomial time, for example in $O(n(\max_i \log a_i)^2)$ bit operations if we use the standard Euclidean algorithm to compute the gcds. (Note that the gcd of n numbers can be computed by computing the gcd of $n-1$ pairs of numbers: $\gcd(a_1, a_2, \ldots, a_n) = \gcd(a_1, \ldots, \gcd(a_{n-2}, \gcd(a_{n-1}, a_n)) \ldots).$)

Next we now show that VERTEX-COVER-DEC $\leq_P$ SUBSET-GCD-DEC. Let $[G = (V, E), k]$ be an instance of vertex cover, $V = \{1, \ldots, n\}$ and $E = \{e_1, \ldots, e_m\}$. Let p_j denote the jth prime: $p_1, p_2, p_3, \ldots = 2, 3, 5, \ldots$. Let P be the product of the first m primes: $P = p_1 p_2 \cdots p_m$. We will associate prime p_j to edge e_j, $1 \leq j \leq m$. For $1 \leq i \leq n$, compute a_i to be the divisor of P equal to the product of all p_j such that e_j is not adjacent to vertex i. Then $V' = \{i_1, \ldots, i_k\}$ is a vertex cover of $G \iff$ for all $p_i \in \{p_1, \ldots, p_m\}$ there exists at least on $j \in V'$ such that p_i does not divide $a_j \iff$ the gcd of the elements in $\{a_{i_1}, \ldots, a_{i_k}\}$ is one. This shows that G has a vertex cover of size at most k if and only if $\{a_1, \ldots, a_n\}$ has a subset of size at most k with gcd equal to one.

Concerning the cost of the construction of $\{a_1, \ldots, a_n\}$, we note that the mth prime has magnitude $O(m \log m)$, so the size of the constructed instance of SUBSET-GCD-DEC will be $O(nm \log(m \log m))$, which is polynomial in the size of $[G, k]$. The first m primes can be found in time polynomial in m by using a naive approach: for $j = 2, 3, 4, \ldots, p_m$ test if j is composite by assaying if j can be expressed as the product of two numbers in the rank $[2, j-1]$.

3. a)

Answer: For a certificate, we can use any convenient representation of the function f. For example, the list Cert $= [f(P_1), f(P_2), \ldots, f(P_t)]$ would work well.
Suppose we are given Cert $= [g_1, \ldots, g_t]$. To verify Cert, we would perform the following checks:

i. Verify that $1 \leq g_j \leq K$ for $1 \leq j \leq t$.
ii. For each i, $1 \leq i \leq s$, verify that there exist $P_j, P_{j'} \in Q_i$ such that $g_j \neq g_{j'}$.

3. b)

The Reduction. We are given an instance of 3-CNF-SATISFIABILITY, i.e., a 3-CNF Boolean formula F in n variables $x_1, \ldots, x_n$, having m clauses $C_1, \ldots, C_m$. We want to construct an instance of GALLERY ALLOCATION, which is defined by a set $\mathcal{P}$, a collection of subsets of $\mathcal{P}$, and an integer $K \geq 2$. Define $\mathcal{P} = \{x_1, \overline{x_1}, \ldots, x_n, \overline{x_n}, z\}$ where z is a new element. For each clause C_i, create a subset $\mathcal{Q}_i$ consisting of the three literals in C_i, along with z. In addition, for each $i = 1, \ldots, n$, we create a subset $\mathcal{Q}_{m+i} = \{x_i, \overline{x_i}\}$. Finally, we choose $K = 2$.

Answer: Suppose I is a yes-instance of 3-CNF-SATISFIABILITY. Denote the constructed instance of GALLERY ALLOCATION by $g(I)$. It is straightforward to show that the construction of $g(I)$ can be done in polynomial time.

We show that $g(I)$ is a yes-instance whenever I is a yes-instance. If I is a yes-instance, then there is a satisfying truth assignment $a : \{x_1, \ldots, x_n\} \rightarrow \{T, F\}$ for the instance I. Define $f : \mathcal{P} \rightarrow \{1, 2\}$ as follows: for every literal $y \in \{x_1, \overline{x_1}, \ldots, x_n, \overline{x_n}\}$, define $f(y) = 1$ if $a(y) = T$ and define $f(y) = 2$ if $a(y) = F$. Also, define $f(z) = 2$. Since a is a satisfying truth assignment, every clause C_i contains $1, 2$ or 3 true literals. The corresponding subset $\mathcal{Q}_i$ therefore contains $1, 2$, or 3 paintings that are assigned to gallery 1. Further, $z \in \mathcal{Q}_i$ is assigned to gallery 2, so $\mathcal{Q}_i$ contains paintings assigned to both galleries. Finally, each set $\mathcal{Q}_{m+i} = \{x_i, \overline{x_i}\}$ contains one painting assigned to gallery 1 and one painting assigned to gallery 2.

Now, assume that $g(I)$ is a yes-instance. Suppose first that $f(z) = 2$. Then define $a : \{x_1, \ldots, x_n\} \rightarrow \{T, F\}$ as follows: for every literal $y \in \{x_1, \overline{x_1}, \ldots, x_n, \overline{x_n}\}$, define $a(y) = T$ if $f(y) = 1$ and define $a(y) = F$ if $f(y) = 2$. Since each set $\mathcal{Q}_{m+i} = \{x_i, \overline{x_i}\}$ contains one painting assigned to gallery 1 and one painting assigned to gallery 2, it follows that $a(x_i) = \text{not}\, a(\overline{x_i})$, so this truth assignment is "consistent". We claim that a is a satisfying truth assignment for the m clauses $C_1, \ldots, C_m$. Let C_i be a clause. The corresponding set $\mathcal{Q}_i$ contains a painting (which is not the painting z) that is assigned to gallery 1; the corresponding literal is a true literal in C_i.

If $f(z) = 1$, then we instead define define $a(y) = F$ if $f(y) = 2$ and define $a(y) = T$ if $f(y) = 1$. The remainder of the proof is similar to the previous case.