2. If any row/column sum not equal k, then it is not expressible by k vectors with exactly one 1.

Let A be an $n \times n$ matrix with nonnegative interger entries, we note that it is equivalent to an weighted bipartite graph $G = (X \cup Y, E; w)$, where $X = \{x_1, ...x_n\}$ and $Y = \{y_1, ..., y_n\}$, and for each non-zero entry $w(x_i y_j) = A[i][j]$. A permutation matrix is equivalent to a perfect matching with every edge weight 1.

We obtain k permutation matrices by repeating the following k times:

1. Find a perfect matching M of G, output the corresponding permutation matrix P.
2. For each edge in M reduce weight by 1 and remove edges with 0 weight.

In i-th iteration, the total weight of edges incident to any vertex is $k - i$. Suppose $S \subseteq X$ and $|S| < |N(X)|$, then some vertex in $N(X)$ must has total weight more than $k - i$, which is a contradiction. By Hall's theorem, there is always a perfect matching.

3. (a) Graph problem: Construct an edge weighted complete graph $G = (V, E; w)$, where each bag is a vertex and the distance between two bags is the edge weight. We assume n is even. (otherwise, add a vertex and connect it to all other vertices with edges of weight 0)

Find a minimum weight perfect matching of G using Edmonds' blossom algorithm in $O(V^2 E)$ time. Then the strategy is that for each pair of matched vertices u, v, load two bags by $door \to u \to v \to door$.

Proof: The worker should carry two bags to the door each time to minimize walking distance. Denote door as D, for any two bags A and B, we have $2AD + 2BD \geq AD + AB + BD$ by triangle inequality.

4. Graph problem: Construct a bipartite graph $G = (X \cup Y; E)$, where $x_1, ..., x_n \in X$ are activities and $y_1, ..., y_m \in Y$ are persons, and an edge $x_i y_j$ if person y_j prefers activity x_i.

Similar to matching, we find a maximum set M of edges such that every x_i incident to at most c_i edges and each y_j incident to at most one. A vertex is M-saturated if it has no remaining capacity and otherwise M-unsaturated.

It is easy to see that M is maximum iff G has no M-augmenting path.

Starting with an arbitrary M, we grow an M-alternating tree T_j for each M-unsaturated vertex $y_j \in Y$ and we can use an M-augmenting path to increase M by one more edge.

Time complexity: grow an alternating tree takes $O(V+E) = O(m)$ and at most $V = m$ trees constructed, it takes at most $O(m^2)$ time.

5. (a) Set $l = 1$ and we get CLIQUE.

(b) Set $k = n$ and we get HAMILTONIAN CYCLE.

(c) Reduce from INDEPENDENT SET. Add a (n+1)-star and connect its center to every vertex $v \in V$. The new graph G' has $k+n+1$ vertices induced tree if and only if G has a independent set of size k.

6. Reduce from VERTEX COVER by replacing each edge $e = uv$ of G with a traingle by adding a new vertex x_e and two edges ux_e, vx_e.

Or add an independent set S of $k+1$ vertices, each is connected so every $v \in G$. If (G,k) has vertex cover X, then $S \cup (G-X)$ is bipartite. On the other hand, at least one vertex $s \in S$ is not removed and if $G-X$ has an edge uv, then uvs is a trianlge.

7. Given an instance (U,C) of 3SAT with n variables and m clauses, we construct a digraph G as follows (the figure is an illustration):

1. For each variable u_i, add two vertices s_i, t_i and two paths of length $m+1$ from s_i to t_i representing u_i and $\bar{u}_i$ respectively;
2. For each clause c_j,add two vertices x_j, y_j and three (x_j, y_j)-paths of length 2 each via a distinct vertex on the path representing its literals;
3. Add edges $t_i s_{i+1}$ for $1 \le i \le n-1$ and $y_j x_{j+1}$ for $1 \le j \le m-1$;
4. Add edges ss_1, sx_1, $t_n t$ and $y_m t$.

Now we show (U,C) is an yes-instance of 3SAT iff G has two vertex disjoint (s,t)-paths of length $l_1 = n(m+2)+1$ and $l_2 = 3m+1$ repectively.

Given a satisfying truth assignment τ, the path of length l_1 consists of ss_1, t_nt, t_is_{i+1} for $1 \leq i \leq n-1$ and for each $1 \leq i \leq n$, if $\tau(u_i) = 1$, the path representing $\bar{u}_i$ and otherwise, the path for u_i. The path of length l_2 consists of sx_1, y_mt, y_jx_{j+1} for $1 \leq j \leq m-1$ and for each clause a path via a literal satisfied by τ.

On the other hand, the longer path must contains one of the entire paths for each variable u_i, assign $\tau(u_i) = 1$ if it uses $\bar{u}_i$ and $\tau(u_i) = 0$ otherwise. The shorter path must contains all vertices x_i and y_i and suppose a clause if unsatisfied by τ, this path will be disconnected.