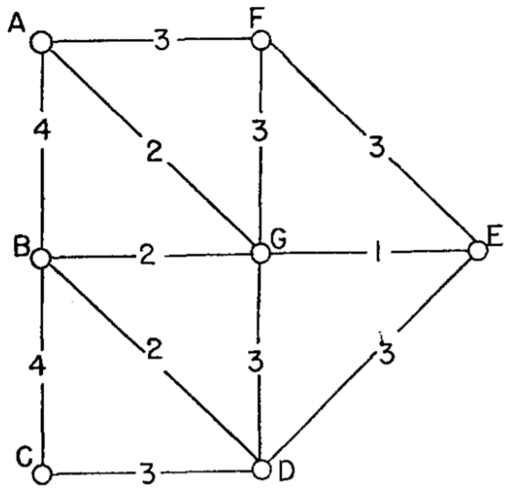
Problem 2 - NP-Completeness & Reduction (Hand-Written) (35 points)

Given a real symmetric matrix $A = (A_{ij})$ of size n, we are interested in finding a permutation matrix P such that $PAP^T = (b_{ij})$ has non-zero entries arranged "near" the diagonal. That is, we want to find the minimal $k \in \mathbb{N}$ such that $b_{ij} = 0$ whenever $|i - j| > k$. A deterministic statement is, given matrix A and natural number $k \in \mathbb{N}$, does there exist a permutation matrix P such that $PAP^T = (b_{ij})$ satisfies $b_{ij} = 0$ for all $|i - j| > k$?

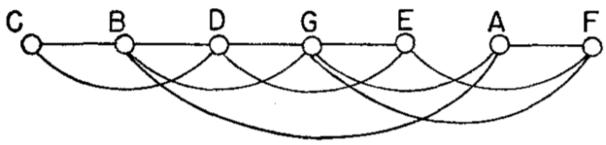
One can consider an undirected graph $G = (V, E)$ that is "induced" by A, where $V = \{v_1, ..., v_n\}$ and $(v_i, v_j) \in E \iff A_{ij} \neq 0$ (therefore, one can regard A as the adjacency matrix of G). We get a graph version of the above question: given a graph G and $k \in \mathbb{N}$, does there exist a 1-1 labeling function τ from V to $\{1, ..., n\}$, such that $\forall (u, v) \in E$, $|\tau(u) - \tau(v)| \leq k$? Formally, given a graph G and $k \in \mathbb{N}$, say a instance of this problem $BMP[G, k]$ is solvable if there exist a labeling function τ from V to $\{1, ..., n\}$, such that $\forall (u, v) \in E$, $|\tau(u) - \tau(v)| \leq k$. This is known as the **Bandwidth Minimization Problem**.

Furthermore, consider a more generalized version:

Given a weighted graph G, say a instance of this problem $LAP[G]$ is solvable if there exist a labeling function τ from V to $\{1, ..., n\}$, such that $\forall (u, v) \in E$, $|\tau(u) - \tau(v)| \leq w(u, v)$. Equivalently, can we arrange the vertices of G in a linear array according to the labeling, such that no two adjacent vertices (in G) have a distance (in the linear array) greater than the weight of the edge joining them? As an instance, the below graph in (a) is solvable by the arrangement (labeling) shown in (b). This is known as the **Linear Array Problem**.



(a)



(b)

*Through problems (a) to (g), we will use the fact the **3-CNF-SAT** problem is NP-complete to prove that the linear array problem is also a NP-complete problem. It is recommended to draw the graphs to understand the problems.*

(a) (4 points) Let S be a set of $2n$ elements, where $n \in \mathbb{N}, n \geq 3$. Define a weighted graph H with $V(H) = S \cup \{M, M'\}$, $E(H) = \{(s, M)|s \in S\} \cup \{(s, M')|s \in S\}$, and let all edges have weight $n+1$ (graph H is shown below). Prove that the problem $LAP[H]$ is solvable, and if τ is any labeling that solves $LAP[H]$, it must satisfy $\{\tau(M), \tau(M')\} = \{n+1, n+2\}$.

Clearly any label τ satisfying $\{\tau(M), \tau(M')\} = \{n+1, n+2\}$ solves $LAP[H]$. Now assume label τ' has $\{\tau(M), \tau(M')\} \neq \{n+1, n+2\}$. WLOG, assume that $\tau(M) <= n$. If $\tau^{-1}(2n+2) = s \neq M'$, then $(s, M) \in E(H)$ but $|\tau(s) - \tau(M)| > n+1$. Otherwise if $\tau(M') = 2n+2$, since $n \geq 3$, there exist $s' \in S$ with $\tau(s') <= n \implies (s', M') \in E(H)$ but $|\tau(s') - \tau(M')| > n+1$. Therefore τ' doesn't solve $LAP[H]$.

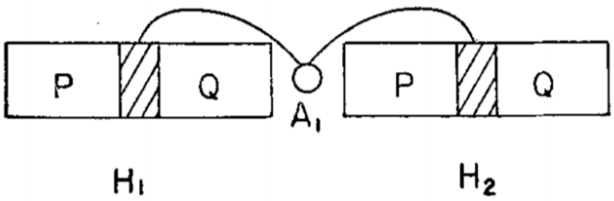
As a consequence, if we denote $P = \{v \in S|\tau(v) < n+1\}$ and $Q = \{v \in S|\tau(v) > n+2\}$, then we have $|P| = |Q| = n$, and P, Q form a partition of S (according to the labeling τ).

Now consider a graph G_1 that contains 2 copies of H introduced in (a) (denoted by H_1, H_2) and an additional vertex A_1, where A_1 is joined with both M, M' of H_1 and H_2 by edges of weight $n+2$.

(b) (4points) Prove that the problem $LAP[G_1]$ described above is solvable, and if τ is any labeling that solves $LAP[G_1]$, it must satisfy $\tau(A_1) = 2n+3$, and $\tau(V(H_1)) = \{1, ..., 2n+2\}$ or $\{2n+4, ..., 4n+5\}$.

Clearly any label τ satisfying $\tau(A_1) = 2n+3$, and $\tau(V(H_1)) = \{1, ..., 2n+2\}$ or $\{2n+4, ..., 4n+5\}$ solves $LAP[H]$. Now assume label τ' solves $LAP[G_1]$. By (a), $\tau(V(H_1)) = \{1+i, ..., 2n+2+i\}$ for some i. If $\tau(A_1) \neq 2n+3$, we must have $|\tau'(A_1) - \tau(v)| > n+2$ where v is one of M or M' in H_1 or H_2, contradiction.

By (b), if we have a labeling τ that solves the problem $LAP[G_1]$, its linear arrangement must be the form showning in the graph below. Therefore, we can define sets P_1, Q_1, P_2, Q_2 according to τ using the method similar to the one in (a). However, note that it is not guaranteed that $P_1 = P_2$.



(c) (5 points) Please add weighted edges to G_1 and get a new graph G_2, such that $LAP[G_2]$ is still solvable, and that if τ is any labeling that solves $LAP[G_2]$, it is guaranteed $P_1 = P_2$. You are required to set the weight of added edges as high as possible. (Hint: for each $s \in S$ in H_1, join it to its corresponding vertex in H_2. Determine the edge weight.)

For each $s \in S$, let $s^{(i)}$ denote the corresponding vertex in H_i. In G_1, join $s^{(1)}$ and $s^{(2)}$ by an edge with weight $2n+5$ to get G_2. Clearly $LAP[G_2]$ is solvable, by choosing τ satisfying the condition in (b) and for each $s \in S$, $\tau(s^{(1)}) = \tau(s^{(2)}) + k$ where $k = \pm(2n+3)$. Now assume τ' solves $LAP[G_2]$ but the induced set P_1, P_2 are unequal. Then there must exist $s \in S$ such that $s^{(1)} \in P_1$ and $s^{(2)} \in Q_2 \implies |\tau(s^{(1)}) - \tau(s^{(2)})| \geq 2n+6$, contradiction.

Quote: In relations "$s^{(1)} \in P_1$ and $s^{(2)} \in Q_2$", $s^{(i)}$ is regarded as an element in S (as P_i, Q_i are regarded as subsets of S). In the formula "$|\tau(s^{(1)}) - \tau(s^{(2)})| \geq 2n+6$", $s^{(i)}$ is regarded as a specific vertex in G_3. The notation $s^{(i)}$ are just for convenience and don't get confused.

(d) (5 points) Given a subset S' of S with $|S'| = 3$. Please add weighted edges to G_2 and get a new graph G_3, such that $LAP[G_3]$ is still solvable, and that if τ is any labeling that solves $LAP[G_3]$, it is guaranteed $S' \cap Q_1$ is non-empty (recall that Q_1 is determined by τ).

Assume $S' = \{x, y, z\} \subseteq S$. In G_2, for each of $x^{(1)}$, $y^{(1)}$ and $z^{(1)}$, join to A_1 with an edge of weight $n + 4$. Call this new graph G_3. Clearly $LAP[G_3]$ is solvable, by choosing τ satisfying the condition in (c) and letting $\{x^{(1)}, y^{(1)}, z^{(1)}\} \subseteq Q_1$. Now assume τ' solves $LAP[G_3]$ but the $S' \cap Q_1 = \phi$, i.e. $S' \subseteq S \setminus Q_1 = P_1$. Then we have $|\tau(v) - \tau(A_1)| > n + 4$ where v is one of $x^{(1)}, y^{(1)}$ and $z^{(1)}$, contradiction.

Now assume that the $2n$ elements of S are $x_1, ..., x_n, y_1, ..., y_n$. Consider a weighted graph G_4 that contains $n + 1$ copies of H and additional vertices $A_1, ..., A_n$, where A_i is joined with both M, M' of H_i and H_{i+1} by edges of weight $n + 2$. Similar to (c), add edges to G_4 such that all partitions of S in $H_1, ..., H_{n+1}$ are the same. Denote P, Q to be this unique partition of S.

(e) (5 points) Please furthermore add edges to G_4 and get a new graph G_5, such that $LAP[G_5]$ is still solvable, and that if τ is a labeling that solves $LAP[G_5]$, it is guaranteed for all $i = 1, ..., n$, there is exactly one of x_i, y_i in P (and the other in Q).
Remark: if the weight of added edges in (c) is too low, G_5 might not be solvable.

In G_4, join $s^{(i)}$ and $s^{(i+1)}$ by an edge with weight $2n+5$ to get G_2 $\forall s \in S, \forall i = 1, ..., n$. Also, join both $x^{(i)}$ and $y^{(i)}$ to A_i with edges of weight $n+3$. Call this new graph G_5. Now consider a labeling τ such that $\tau(H_i) = \{(i-1)(2n+3)+1, ..., (i-1)(2n+3)+(2n+2)\}$ for $i = 1, ..., n+1$, $\tau(A_j) = j \times (2n+3)$ for $j = 1, ..., n$, and the vertices in H_i are $y_i, y_{i+1}, ..., y_{i+n-1}, M, M', x_i, x_{i+1}, ..., x_{i+n-1}$ of descending order of labels (let $x_j := x_{j-n}$ if $j > n$. same for y_j). One can check that this label indeed solves $LAP[G_5]$. Now assume that τ' solves $LAP[G_5]$ but $\exists j$ such that both $x_j, y_j \in Q \Longrightarrow$ we must have $|\tau(x_j^{(j)}) - \tau(A_j)| > n + 3$ or $|\tau(y_j^{(j)}) - \tau(A_j)| > n + 3$, contradiction.

Finally, we can combine the results in (a) to (e) and prove that problem LAP is NP-complete:

(f) (8 points) Consider an instance B of ***3-CNF-SAT*** with literals $x_1, ..., x_n$ and clauses $F_1, ..., F_r$. Construct a weighted graph G which contains $n+r+1$ copies of H and nodes $A_1, ..., A_{n+r}$, with some additional edges, such that $LAP[G]$ is solvable $\iff$ B is a satisfiable boolean formula.

(Hint: let the first n copies of H force S become consistent partition P, Q, and the next r copies of H force the partition $P \cup Q$ to satisfy B, assuming the literals in Q are given the value 1, and the literals in P are given the value 0.)

Let $S = \{x_1, ..., x_n, \neg x_1, ..., \neg x_n\}$ for $n \geq 3$. Let G contain $H_1, ..., H_{n+r+1}$ and additional vertices $A_1, ..., A_{n+r}$, where A_i is joined with both M, M' of H_i and H_{i+1} by edges of weight $n + 2$.

Join $s^{(i)}$ and $s^{(i+1)}$ by an edge with weight $2n + 5$ to get G_2 $\forall s \in S, \forall i = 1, ..., n + r$. Also, join both $x^{(i)}$ and $y^{(i)}$ to A_i with edges of weight $n + 3$ for $i = 1, ..., n$.

Let $S_i = \{s_1^{(i)}, s_2^{(i)}, s_3^{(i)}\}$ be the set of 3 literals occurring in F_i ($s_j^{(i)} = x_k$ or $\neg x_k$ for some k). At last, for each $j = 1, ..., r$, connect A_{n+j} to all vertices in S_j with edges of weight $n + 4$.

($\Longrightarrow$)

Now suppose $LAP[G]$ is solved by labeling τ. By (c), the partitions P_i, Q_i are all the same. Denote P, Q to be this unique partition of S. By (e), the partition is consistent, i.e. for all $i = 1, ..., n$, there is exactly one of $x_i, \neg x_i$ in P (and the other in Q). Now consider the assignment that the literals in P are given the value 0, the literals in Q are given the value 1. By (d), each $S_j \cap Q$ is non-empty $\Longrightarrow$ F_j is satisfied for all $j = 1, ..., r \Longrightarrow B$ is satisfied by this assignment.

($\Longleftarrow$)

Conversely, assume B is satisfied by an assignment. Suppose that $z_1, ..., z_n$ are the literals assigned the value 0, where $z_i = x_i$ or $\neg x_i$. Let S_i be defined same as above, and assume that $s'_{i,1}, ..., s'_{i,p_i}$ are the literals in S_i that are assigned the value 0, where $p_i = 0$ or 1 or 2.

Let τ be a labeling, such that $\tau(H_i) = \{(i-1)(2n+3)+1, ..., (i-1)(2n+3)+(2n+2)\}$ for $i = 1, ..., n+r+1$, $\tau(A_j) = j \times (2n+3)$ for $j = 1, ..., n+r$. Note that if we want τ to solve $LAP[G]$, we must assign z_j to be the vertex of largest label in H_j for $j = 1, ..., n$ (to satisfy the length limit set by added edges of (e)), and assign $s'_{i,1}, ..., s'_{i,p_i}$ to be the vertices or largest labels in H_{n+i} for $i = 1, ..., r$ (to satisfy the length limit set by added edges of (d)).

Now consider a qeque D, which has elements $z_1, ..., z_n$ in order at round 1. At round $1 < i \leq n+r+1$, pop "literals that must have largest label in P_{i-1}" (which is z_i for $i \leq n$, and are $s'_{i,1}, ..., s'_{i,p_i}$ for $i > n$), and push "literals that must have largest label in P_{n+i-1}" (add no literals of $i > r+2$). Set the label of vertices in P_i according to D at round i: from the front to the bottom of D, assign literals to have label $(i-1)(2n+3)+(2n+2), (i-1)(2n+3)+(2n+1), ..., (i-1)(2n+3)+(s+1)$ in order(if an literal exists more than 1 time, only set its label at the first time and ignore the rest); for the literals $z_{q_1}, z_{q_2}, ...z_{q_s}$ in P that are not in D in round i, assign literals to have label $(i-1)(2n+3)+s, (i-1)(2n+3)+(s-1), ..., (i-1)(2n+3)+1$ in order of the size of index q_j. One can easily check that under this assignment, the corresponding literals in P_i and P_{i+1} has length no more than $2n+5$. At last, set $\neg z_j^{(j)}$ (in Q_i) to have label $\tau(z_j^{(i)}) + n + 1$ for all $i = 1, ..., n+1, j = 1, ..., n$. Then this is a labeling that solves $LAP[G]$.

(g) (4 points) By the fact that ***3-CNF-SAT*** is NP-complete, conclude that problem LAP is NP-complete. You are required to show that the reduction is indeed in polynomial time.

Let G be an instance of LAP and $|V(G)| = N$. To verify whether $LAP[G]$ is solvable, there are $O(N^2)$ edges to check if $|\tau(u) - \tau(v)| \leq w(u, v)$, clearly can be done in polynomial time. Thus $LAP \in$ NP.

On the other hand, (f) gives an reduction algorithm to reduce any instance of *3-CNF-SAT* to an instance of problem LAP. Notice that the graph has $O(n(n+r))$ vertices and $O(n^2(n+r)^2)$ edges, thus clearly the reduction is in polynomial time. By the fact that *3-CNF-SAT* $\in$ NP-hard, we have $LAP \in$ NP-hard.

Combining the above results and we have $LAP \in$ NP-complete.

We finally proved that the **Linear Array Problem** is NP-Complete. However, the above argument does not imply that the **Bandwidth Minimization Problem** is NP-complete. To prove that the **Bandwidth Minimization Problem** is NP-complete, we modify the construction of G in (f), instead we get a graph G' such that all edges have weight b or $2b-1$ for some $b \in \mathbb{N}$, and that such that $LAP[G']$ is solvable $\iff$ F is a satisfiable boolean formula. After that, reduce this "restricted" problem $LAP[G']$ to problem $BMP[G'', b']$. To learn the details, you are encouraged to read this paper.