We are given a directed and weighted graph $G = (V, E, w)$, where $w(e) > 0$ for each $e \in E$. Additionally, each edge is coloured either red or blue.

(a) Let $\textsc{onlyRedEdges}(u, v, \ell)$ denote the weight of the shortest path from $u$ to $v$ that uses *only* red edges and have length at most $\ell$. For a fixed pair of points $u$ and $v$, describe an $O(n^3 \log k)$ algorithm to compute $\textsc{onlyRedEdges}(u, v, k-1)$.

**Hint.** *You should only need $O(n^2 \log k)$ subproblems.*

**Hint 2.** *Alternatively... you can just apply Bellman-Ford.*

(b) Hence, describe an $O(n^3 \log k)$ algorithm that returns a *closed walk* with the smallest weight that contains *at least* one blue edge and does not contain $k$ consecutive red edges.

A *closed walk* is a walk that starts and ends at the same vertex.

**Solution.**

(a) We initialise a Dynamic Programming table $DP$ with $DP[i][j]$ to represent the shortest path using only red edges from vertex $u$ to $j$ using at most $i$ edges. Obviously, $DP[0][u] = 0$. Now, for all $i$ from $1$ to $k-1$, we take each vertex $j$ and set $DP[i][j] = DP[i-1][j]$, then for each vertex $v$, if there is a red edge $(v, j)$, we update $DP[i][j] = \min\{DP[i][j], DP[i-1][v] + w(v, j)\}$.

(b) Solution to part (b) goes here...