1 Question 1

a)

Prove that G has a clique of size $t \iff G^k$ has a clique of size t^k.

($\Rightarrow$) Let C be a clique of size t in G. Denote the vertices in the clique as $c_i \in C$, for all integer $i \in [1, t]$. Consider C', the subset of vertices in G^k where for each $(v_1, \dots, v_k), v_i \in C$. Now, for any two vertices $(u_1, \dots, u_k)$ and $(v_1, \dots, v_k)$ in C', we see that for all i, either $u_i = v_i$ or there must be an edge between u_i and v_i in G as $u_i, v_i \in C$. Thus, all vertices in C' are completely connected to each other in G^k, meaning that C' is a clique. Observe that for each of the k entries in each tuple-vertex of C', we can choose any of the t vertices from C. Thus, since there are t^k possible combinations for tuples of size k, we have $|C'| = t^k$. Thus, a clique of size t^k exists in G^k.

($\Leftarrow$) Following from the same conventions as above, C' is a clique of size t^k in G^k. This must mean that there are t^k distinct vertices in C', which must mean that there are t^k distinct combinations of $(v_1, \dots, v_k) \in C'$. To get t^k distinct combinations, you need at least t distinct elements for at least one index of the k-tuples in C' (i.e. suppose you have $t - 1$ distinct elements, then the most number of unique k-tuples you can make is $(t - 1)^k$). Since all vertices are completely connected in C', we know that for the index that has at least t distinct possible values/vertices, there must be an edge in G between each pair of vertices/values for that index - otherwise there wouldn't be an edge between the k-tuple vertices in C'. Thus, there are at least t vertices that are completely connected to each other in G, meaning there is a clique of size t in G.

b)

Intuition

Suppose the maximum clique in G has size t. "Raising" G to some k will map this clique to a larger clique of size t^k. A constant factor approximation algorithm will be able to find a clique that is a subset of these vertices, say of size $\frac{t^k}{2}$. We know that this clique has $\frac{t^k}{2}$ unique combinations of k-tuples. Correspondingly, for some index of the clique of k-tuples, there must be at least $\sqrt[k]{\frac{t^k}{2}}$ distinct values for $\frac{t^k}{2}$ combinations to be possible. Since the clique is completely connected, we know that those $\sqrt[k]{\frac{t^k}{2}}$ vertices are completely connected in G and are must form a clique. Note that the clique found in G approximates to $\frac{t}{\sqrt[k]{2}}$ - this means as $k \to \infty$, we have $\frac{t}{\sqrt[k]{2}} \to t$.

Proof

Given G, let's find the maximum cover in G^k for some k. Let OPT be the maximum cover in G and OPT^k be the maximum cover in G^k.We will determine this k later (honestly, what's with determining everything retrospectively). Run A on G^k to find a cover C^k. We know that $|C^k| \geq \frac{1}{2}|OPT^k|$ because the approximation factor of A is $\frac{1}{2}$. Let $m \geq \frac{1}{2}$ be the actual approximation achieved - thus, $|C^k| = m|OPT^k|$. Now, going through the k-tuples $\in$C^k, we have at least $|C^k| = t^k$ possible combinations for some t. By the previous part, we know that there is a clique in G of size t, which is an approximation of the maximum clique. Let $t = (1-\epsilon)|OPT|$. Thus, we have $((1 - epsilon)|OPT|)^k = |C^k| = m|OPT^k|$.

Note: In practice, we know that at least one index in the k-tuples in C^k have $\lceil t \rceil$ distinct values (otherwise, t^k combinations are not possible for C^k).

Observe that $|OPT|^k = |OPT^k|$ as the largest clique in G maps to a clique in G' and vice versa by the above question. Now, let's re-arrange:

$$
\begin{aligned}
((1-\epsilon)|OPT|)^k &= m|OPT^k| \\
(1-\epsilon)^k|OPT|^k &= m|OPT^k| \\
(1-\epsilon)^k &= m \\
k\log(1-\epsilon) &= \log m \\
k &= \frac{\log m}{\log(1-\epsilon)} \\
k &= \frac{-1}{\log(1-\epsilon)} \quad \text{worst case } m = \frac{1}{2}\text{; greater } m \text{ requires smaller } k
\end{aligned}
$$

Thus, for some given ϵ such that we get an approximation clique of size $(1 - \epsilon)|OPT|$ in G, we need to at least $k = \frac{-1}{\log(1-\epsilon)}$.

Algorithm

Based on the above proof, we first determine the necessary $k = \frac{-1}{\log(1-\epsilon)}$ based on the given ϵ. Then, we create G^k. We run A on G^k and then for the clique C^k returned, we look through every index to find the largest list of values/vertices possible for that index. The maximal list will be C such that $|C| \geq (1-\epsilon)|OPT|$.

Analysis

The bottle-neck of this algorithm is generating G^k. For $k = \frac{-1}{\log(1-\epsilon)}$. It's clear that there are n^k vertices (n choices for each of the k indices in the tuple).

Algorithm 1 PTAS for Clique

procedure PTASClique($G = (V, E)$, ϵ)
    $k \leftarrow \dfrac{-1}{\log(1-\epsilon)}$
    create graph G^k brute-force
    $C^k \leftarrow \mathrm{A}(G^k)$
    $C \leftarrow \emptyset$
    for $i \in [1, \ldots, |C^k|]$ **do**
        $S \leftarrow$ distinct vertices in C^k for index i
        if $|S| \geq |C|$ **then**
            $C \leftarrow S$
            return C

There are at most $O(n^{2k})$ edges. A brute force graph construction of G^k in which you iterate over every vertex/k-tuple, for each index and compare it with all the other vertices is bounded by $O(n^{3k})$. Running A on this graph is $O(p(n^k))$. Iterating through the maximal clique in G^k to find the maximal list of vertices for each index is no more than $O(n^{3k})$ (high upper bound than necessary because it's irrelevant/not the bottleneck). Thus, we see that the total runtime of the algorithm is $O(n^{\frac{-2}{\log(1-\epsilon)}} + p(n^{\frac{-1}{\log(1-\epsilon)}}))$.