1. (a) Consider the complete graph $H(G) = (V \cup E, (V \cup E) \times (V \cup E))$ and use the following weights:

$$\begin{array}{ll} \forall u, v \in V & W(u,v) = 1 \\ \forall e, f \in E & W(e,f) = |E| + |V| \\ \forall v \in V, e \in E & W(v,e) = \begin{cases} 1 & v \in e \\ |E| + |V| & v \notin e \end{cases} \end{array}$$

Let $R = E$ and $q = |E| + k - 1$. Obviously the reduction is polynomial.

Proposition. $H(G)$ has a Steiner Tree T spanning R of weight $\leq q$ if and only if G has a VC $S \subset V$ of size $\leq k$.

Proof. ($\Leftarrow$) Consider the subgraph of H consisting of $E \cup S$ and the edges of weight 1 between them. S is a VC, so the subgraph is connected, hence contains a spanning tree T. $|E \cup S| = |E| + |S| \leq |E| + k = q + 1$, so the T weighs $\leq q$.

($\Rightarrow$) Any tree of weight $\leq q$ cannot use any of the heavy ($= |E| + |V|$) edges, so T only uses edges of weight 1. Consider the set $S \subset V \cap T$. T spans E, so every $e \in E$ must have an edge (of weight 1) to some vertex in S, so S is a VC in G. But T has $\leq q$ edges and $|E|$ of them connect E to S; that leaves $\leq q - |E| = k - 1$ edges between vertices of S, so $|S| \leq k$. □

(b) Consider the complete graph $H'(G) = (V \cup E \cup \{r\}, (V \cup E \cup \{r\}) \times (V \cup E \cup \{r\}))$ and use the following weights:

$$\begin{array}{ll} \forall u, v \in V & W(u,v) = 2 \\ \forall e, f \in E & W(e,f) = 2 \\ \forall v \in V, e \in E & W(v,e) = \begin{cases} 1 & v \in e \\ 2 & v \notin e \end{cases} \\ \forall v \in V & W(r,v) = 1 \\ \forall e \in E & W(r,e) = 2 \end{array}$$

Let $R = E \cup \{r\}$ and $q = |E| + k$. Obviously the reduction is polynomial.

Note that since all weights are 1 or 2, the triangle inequality holds for W.

Proposition. $H'(G)$ has a Steiner Tree T spanning R of weight $\leq q$ if and only if G has a VC $S \subset V$ of size $\leq k$.

Proof. ($\Leftarrow$) Same as in 1a, except the subtree inside V is replaced by a star whose origin is r.

($\Rightarrow$) Consider the Steiner Tree T under weight $\leq q$ spanning R that has the least number edges of weight 2. We claim that this number is 0: assume that T is rooted ar r and consider the path from $e \in E$ via an endpoint $v \in e$ to r. This path uses two edges of weight 1, so it can replace any path of weight ≥ 2.

From here the proof continues as in 1a with the obvious adjustments. □

(c) Using the reduction from 1b we can show that

$$\textsc{Gap}_{[|V|\alpha,|V|\beta]}\textsc{Bounded-Degree-VC} \le_P \textsc{Gap}_{[|V|\alpha+|E|,|V|\beta+|E|]}\textsc{Metric-ST}$$

So it is NP-hard to approximate within

$$\frac{|V|\beta+|E|}{|V|\alpha+|E|} = 1 + \frac{|V|(\beta-\alpha)}{|V|\alpha+|E|} \ge 1 + \frac{|V|(\beta-\alpha)}{|V|\alpha+2|V|\Delta} = 1 + \frac{\beta-\alpha}{\alpha+2\Delta} = \frac{\beta+2\Delta}{\alpha+2\Delta} > 1$$

2. Assume without loss of generality that for every variable x, both the literal x and $\neg x$ occur in ϕ since otherwise we can substitute $x = true/false$ in ϕ and drop the clause(s) in which it appears, without affecting the satisfiablity of ϕ.

Consider now the clause-variable incidence bipartite graph $H(\phi)$ with vertex sets C (clauses) and X (variables).

Proposition. ϕ is satisfiable if and only if $H(\phi)$ has a matching that saturates C.

Proof. ($\Rightarrow$) Every truth assignment to the variable $x \in X$ satisfies exactly one clause; thus, in order to satisfy all clauses in $C' \subseteq C$, we have to have $|C'|$ variables in $N(C')$. But ϕ is satisfiable, so for all $C' \subseteq C$, $|N(C')| \ge |C'|$. The result follows by Hall's Theorem.

($\Leftarrow$) Every clause $c \in C$ has a unique matching variable $x \in X$. Assign to x the value that will satisfy c. Now all clauses have been satisfied. $\square$

H can be constructed in polynomial time and a saturating matching can be calculated in polynomial time using max-flow algorithms, so 2-Occ-3-SAT $\in P$.

3. (a) Note that for every cut $S \subset V$, $E = E(S,S) \cup E(V-S,V-S) \cup E(S,V-S)$; thus a cut is maximal if and only if the uncut is minimal. We've shown in class that $\textsc{Gap}_{[\alpha,\beta]}$Max-Cut is NP-hard for some $0 < \alpha < \beta < 1$, so the trivial reduction shows that $\textsc{Gap}_{[1-\beta,1-\alpha]}$Min-Uncut is NP-hard as well. Therefore, it is NP-hard to approximate within $c = \frac{1-\alpha}{1-\beta} > 1$.

   (b) Either use a direct probablistic argument – $\mathrm{E}(|E(S,S) \cup E(V-S,V-S)|) = \frac{|E|}{2}$ – or remember that every graph contains a cut of size $\ge \frac{|E|}{2}$, so the corresponding uncut is of size $\le \frac{|E|}{2}$.

   (c) Use the 2-approximation algorithm for Max-Cut shown in class.

   (d) No. For instance, a non-complete bipartite graph always has a uncut of size 0, but the algorithm might find a larger uncut (e.g., for a path of length 3, the algorithm might put both ends in S).

4. (a)

(b)

$$\begin{aligned} PSPACE &\subseteq P^{PSPACE} && \text{since} && \text{we can pass the input to the oracle and return its reply.} \\ &\subseteq NP^{PSPACE} && \text{since} && P \subseteq NP \\ &\subseteq PSPACE^{PSPACE} && \text{since} && NP \subseteq PSPACE \\ &= PSPACE && \text{since} && \text{we can process the queries ourselves in polynomial space.} \end{aligned}$$

Thus $P^{PSPACE} = PSPACE \overset{\text{by 4c}}{=} NPSPACE = NP^{PSPACE}$.

(c) Obviously $PSPACE \subseteq NPSPACE$; the other direction is a result of Savitch Theorem, as $NSPACE(p(n)) \subseteq SPACE(p^2(n))$ for all polynomials $p(n)$.

(d)

(e)

$$\begin{aligned} EXP &\subseteq P^{EXP} && \text{since} && \text{we can pass the input to the oracle and return its reply.} \\ &\subseteq NP^{EXP} && \text{since} && P \subseteq NP \\ &\subseteq EXP && \text{since} && \text{we can try all guesses and process each of the polynomially sized queries in exponential time.} \end{aligned}$$

Thus $P^{EXP} = NP^{EXP} = EXP$.

Note: we've shown here $NP^{EXP} = EXP$; we shouldn't hope for a stronger claim unless we solve the open question $EXP \overset{?}{=} NEXP$.