A.A. Σ^* for any non-empty alphabet Σ.

A.B. $\{0^m 1^m : m \geq 0\}$.

A.C. $\{0^m 1^m 2^m : m \geq 0\}$.

A.D. $\mathrm{ACC}_{\mathrm{TM}}$, $\mathrm{HALT}_{\mathrm{TM}}$, etc.

A.E. We don't know, but if any language fits the criteria, then it's an NP-complete language: SAT, MAX-CUT, etc. These languages are elements of P if and only if P = NP.

A.F. There is no such language, because PSPACE $\subseteq$ NPSPACE, because deterministic Turing machines can be regarded as a special case of non-deterministic Turing machines.

B. From basic facts and Savitch's theorem, we know that

$$\mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{NPSPACE} = \mathrm{PSPACE}.$$

If P = PSPACE, then P = NP (which is a big outstanding problem), and NP = NPSPACE (which isn't currently known either, although we haven't talked about that much). Conversely, if P $\neq$ PSPACE, then either P $\neq$ NP or NP $\neq$ NPSPACE (or both). So no, we don't know whether P = PSPACE, and finding the answer to this question would be a major event in theoretical computer science.

C. An input $\langle G, k\rangle$ is in MAX-CUT if G is an undirected graph that has a cut of size at least k. (A cut is a partitioning of G's nodes into two "sides", and the size of the cut is the number of edges that cross from one side to the other.) The verifier V would be given not just $\langle G, k\rangle$ but also a certificate c, which could be a list of nodes in G. It would take these nodes to constitute one side of the cut and all unlisted nodes to constitute the other side. It would then count the number of edges crossing from one side to the other, and check that it is at least k. This could all be done in time polynomial in the size of $\langle G, k\rangle$.

D.A. The language of D consists of all $\langle M, N\rangle$ such that $L(M) = L(N)$. To see so, consider the product DFA P whose final states are

$$F_P = F_M \times (Q_N - F_N) \cup (Q_M - F_M) \times F_N.$$

That is, P accepts an input w if and only if M and N disagree about w. If $L(P)$ contains a string w of length greater than or equal to $p = |Q_P|$, then w can be pumped down (repeatedly, if necessary) to obtain a string $x \in L(P)$ of length less than p. In other words, if P rejects all strings of length less than (or equal to) p, then it rejects all strings, so $L(P) = \emptyset$ and $L(M) = L(N)$.

D.B. Both $|Q_M|$ and $|Q_N|$ are polynomially related to the size of the input $\langle M, N\rangle$. So their product p is too. And p is the amount of space needed to store each string x, so it is approximately the amount of extra working space needed by D. Thus the space complexity of D is polynomial.

E. [This was a difficult problem, but students were given it ahead of time as a practice problem, so I felt that it was reasonable. Also, I made it worth 8 points instead of 12.] Suppose for the sake of contradiction that a TM R recognizes A. We design a recognizer S for $\text{HALT}^c_{\text{TM}}$ as follows. On input $\langle M, w\rangle$, S does these steps:

1. Build a TM N that, on input x, does these steps:
   (a) Run M on input w for $|x|^2$ steps.
   (b) If M halts in that allotted time, then enter into an infinite loop.
   (c) If M does not halt in that allotted time, then accept.
2. Run R on $\langle N\rangle$ and output whatever R outputs.

To see that S recognizes $\text{HALT}^c_{\text{TM}}$, first suppose that $\langle M, w\rangle \in \text{HALT}^c_{\text{TM}}$. Then

$$\begin{aligned}
\langle M, w\rangle \in \text{HALT}^c_{\text{TM}} &\Leftrightarrow M \text{ loops on input } w \\
&\Rightarrow N \text{ accepts every input } x \text{ in time } \mathcal{O}(|x|^2) \\
&\Rightarrow N \text{ halts on all inputs in time } \mathcal{O}(n^2) \\
&\Leftrightarrow R \text{ accepts } \langle N\rangle \\
&\Leftrightarrow S \text{ accepts } \langle M, w\rangle.
\end{aligned}$$

Second, suppose that $\langle M, w\rangle \notin \text{HALT}^c_{\text{TM}}$. Therefore M halts on w in some amount t of time. Let x be any string such that $|x|^2 \geq t$. Then N loops on x, and

$$\begin{aligned}
N \text{ loops on } x &\Rightarrow N \text{ does not halt on all inputs in time } \mathcal{O}(n^2) \\
&\Leftrightarrow R \text{ rejects or loops on } \langle N\rangle \\
&\Leftrightarrow S \text{ rejects or loops on } \langle M, w\rangle.
\end{aligned}$$

Therefore S is a recognizer for $\text{HALT}^c_{\text{TM}}$. However, $\text{HALT}^c_{\text{TM}}$ is not recognizable. This contradiction implies that A is also not recognizable.