1. (SOLO) Which of the following languages are regular? If you think a language is regular, draw the state diagram of a DFA that recognizes the language. You do not have to prove that the DFA recognizes the language. If you think a language is not regular, than you have to provide a proof.

   (a) $L = \{ a^n b^m \mid n, m \geq 1, n \equiv m \, \text{mod} \, 3 \}$, where $\Sigma = \{a, b\}$.
   (b) $L = \{ w \in \Sigma^* \mid w \text{ contains at least two } a's \text{ and at most one } b \}$, where $\Sigma = \{a, b\}$.
   (c) $L \subseteq \{a, b\}^*$ is the set of all strings whose second-to-last character is a $b$.
   (d) $L = \{ xay : x, y \in \{a, b\}^*, |x| = |y| \}$, where $\Sigma = \{a, b\}$. ($L$ is the set of strings with an $a$ at the center.)
   (e) $L = \{ w0^n w : w \in \Sigma^*, |w| \geq 1, n \geq 1 \}$, where $\Sigma = \{0, 1\}$.
   (f) $L = \{ 0^n w0^n : w \in \Sigma^*, |w| \geq 1, n \geq 1 \}$, where $\Sigma = \{0, 1\}$.

2. (SOLO) Let $\Sigma = \{a, b\}$.

   (a) Prove that any DFA recognizing $L = \{ a^n b^m : n, m \in \mathbb{N} \}$ must have at least 3 states.
   (b) Let $K \subseteq \Sigma^*$ denote the set of all strings that contain $abb$ or $aab$ as a substring. Show that any DFA recognizing $K$ must have at least 5 states. (You may want to start by drawing a DFA with 5 states that recognizes $K$.)

3. (SOLO)

   (a) Let $\Sigma$ be an alphabet that contains the symbol #. Given a language $L \subseteq \Sigma^*$, define $FILL(L) = \{ u\#v : uv \in L \}$.

   Show that if $L$ is regular, then $FILL(L)$ must also be regular, using the threading idea seen in class (make sure you understand really well the proof that regular languages are closed under concatenation before attempting this question). In particular, give an exact description of a DFA recognizing $FILL(L)$, explicitly stating how $Q$, $\delta$, $q_0$ and $F$ are defined. Furthermore, briefly explain the reasoning behind your construction to help the reader understand it. A detailed proof of correctness is not needed.

   (b) Fix an alphabet $\Sigma$. If $x = a_1 a_2 \ldots a_n$ and $y = b_1 b_2 \ldots b_n$ are two strings of the same length, define $mix(x, y)$ to be the string in which the symbols of $x$ and $y$ alternate, starting with the first symbol of $x$, that is, $mix(x, y) = a_1 b_1 a_2 b_2 \ldots a_n b_n$. If $L_1$ and $L_2$ are languages, define $MIX(L_1, L_2)$ to be the language of all strings of the form $mix(x, y)$, where $x$ is any string in $L_1$ and $y$ is any string in $L_2$ of the same length. Prove that if $L_1$ and $L_2$ are regular, then so is $MIX(L_1, L_2)$. You should give an exact description of a DFA recognizing $MIX(L_1, L_2)$, explicitly stating how $Q$, $\delta$, $q_0$ and $F$ are defined. Once you have the exact definition, give a short explanation why this DFA recognizes $MIX(L_1, L_2)$. A detailed proof of correctness is not needed.
4. (GROUP) Let $\Sigma = \{a, b\}$ and define $f : \Sigma \to \Sigma^*$ such that $f(a) = ab$ and $f(b) = a$. Then $f^* : \Sigma^* \to \Sigma^*$ is defined as follows: $w = \sigma_1 \sigma_2 \ldots \sigma_k \in \Sigma^*$, $f^*(w) = f(\sigma_1)f(\sigma_2)\ldots f(\sigma_k)$. We now define recursively a language $L \subseteq \Sigma^*$ as follows:

- $a \in L$;
- if $w \in L$, then $f^*(w) \in L$.

For $n \in \mathbb{N}$, let $w_n \in \Sigma^*$ be the word generated after $n$ applications of the recursive rule above. So $w_0 = a$, $w_1 = ab$, $w_2 = aba$, and so on. Prove that $|w_n| = F_{n+1}$, where $F_n$ is the $n$'th Fibonacci number. Recall that $F_0 = 1$, $F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$.

5. (GROUP) There are 251 Bad Guys and 251 Good Guys. The 251 Bad Guys line themselves up in a room, each carrying a white hat and a black hat. One by one, Good Guys come into the room. When Good Guy $i$ comes in ($1 \leq i \leq 250$), one of the hatless Bad Guys waves his hand. (The Bad Guys get to choose who waves.) Good Guy $i$ then gets to tell the waving Bad Guy to put on either his black hat or his white hat. Following this, the Good Guy leaves, never to be seen again (i.e., he cannot communicate with the future incoming Good Guys).

After the first 250 Good Guys have entered the room the game changes. At this point, only one Bad Guy is hatless—let’s call him the Bad Captain. The Bad Captain must put on a hat now, but he gets to decide the color. Finally, the last Good Guy enters the room—let’s call him the Good Captain. In some sense, his task is to guess—based on the hat colors he sees—who the Bad Captain is. More precisely, the Good Captain must announce some subset $J$ of the Bad Guys, and $J$ must be guaranteed to contain the Bad Captain. The Good Captain then pays a price of $|J|$ dollars. For example, the Good Captain is allowed to pick $J$ to be all 251 Bad Guys. However, this costs a lot: $\$251$. It’s possible for the Good Guys to do better.

The Good Guys are trying to use a strategy that guarantees they’ll never pay more than some $C$ dollars. (They can decide on a strategy together before the game starts.) The Bad Guys are trying to use a strategy that forces the Good Guys to pay as much as possible.

Find and prove a strategy for the Good Guys that keeps their cost $C$ as small as you can. You certainly do not have to prove your $C$ is as small as possible (in fact, we personally don’t know how to do this). You should just strive to get $C$ small. Getting it in the low triple-digits is a reasonable start. Getting it in the low double-digits would be a lot better.

(Remark: this problem doesn’t have anything to do with what we covered in class. It’s basically just a puzzle. However, it is characteristic of some of the types of problems that actually arise in theoretical computer science. Indeed, at a mystery date in the future we will describe how this exact problem arose...)

6. (OPEN) Fix $\Sigma = \{a\}$. Prove that $\{a^{2n} : n \in \mathbb{N}\} \subseteq \Sigma^*$ is not a super awesome language. Note that you may not use the fact that super awesome languages are exactly the same as regular languages, because we did not prove this highly non-trivial fact. Your proof should not make any references to the concept of DFAs. What you have learned in the first week is sufficient to solve this problem.

7. (BONUS - SOLO) For a language $L \subseteq \Sigma^*$, define

$$\text{HALF}(L) = \{x \in \Sigma^* : xx \in L\}.$$  

Prove that if $L$ is regular, then so is $\text{HALF}(L)$.  