1. More on Boolean function forms
   a. Boolean function from last time
      i. True when either, but not both, of the first two variables is true
      ii. Truth table below

Index	A	B	C	f(A, B, C)	Minterm	Maxterm
0	0	0	0	0	$m_0 = \overline{ABC}$	$M_0 = A + B + C$
1	0	0	1	0	$m_1 = \overline{AB}C$	$M_1 = A + B + \overline{C}$
2	0	1	0	1	$m_2 = \overline{A}B\overline{C}$	$M_2 = A + \overline{B} + C$
3	0	1	1	1	$m_3 = \overline{A}BC$	$M_3 = A + \overline{B} + \overline{C}$
4	1	0	0	1	$m_4 = A\overline{BC}$	$M_4 = \overline{A} + B + C$
5	1	0	1	1	$m_5 = A\overline{B}C$	$M_5 = \overline{A} + B + \overline{C}$
6	1	1	0	0	$m_6 = AB\overline{C}$	$M_6 = \overline{A} + \overline{B} + C$
7	1	1	1	0	$m_7 = ABC$	$M_7 = \overline{A} + \overline{B} + \overline{C}$

   b. Sum-of-products
      i. f = $\overline{A}B\overline{C} + \overline{A}BC + A\overline{BC} + A\overline{B}C$ = $m_2 + m_3 + m_4 + m_5$
      ii. Can simplify using equivalence laws to reduce number of gates
         1. f = $\overline{A}B\overline{C} + \overline{A}BC + A\overline{BC} + A\overline{B}C$
         2. = $\overline{A}B(\overline{C} + C) + A\overline{B}(\overline{C} + C)$ by distributive law
         3. = $\overline{A}B + A\overline{B}$ by OR complement law
   c. Product-of-sums
      i. f = $(A + B + C) * (A + B + \overline{C}) * \overline{(A} + \overline{B} + C) * (\overline{A} + \overline{B} + \overline{C})$ = $M_0 * M_1 * M_6 * M_7$
      ii. Can also simplify using laws of equivalence
         1. f = $(A + B + C) * (A + B + \overline{C}) * (\overline{A} + \overline{B} + C) * (\overline{A} + \overline{B} + \overline{C})$
         2. = $\left((A + B)(C + \overline{C})\right) * ((\overline{A} + \overline{B})(C + \overline{C}))$ by distributive law
         3. = (A + B) * $(\overline{A} + \overline{B})$ by OR complement law

2. Karnaugh maps
   a. Method of simplifying Boolean expressions visually
   b. Hamming distance – minimum number of bits needed to change a binary number to another number
      i. 101 to 110 has a Hamming distance of two
   c. Karnaugh maps have a Hamming distance of one between all adjacent cells
      i. Why the order on a 2-variable row/column goes 00, 01, 11, 10, see next page
   d. Map wraps around top, bottom, sides
   e. Place a 1 wherever the function is true, 0 elsewhere
   f. Examples on next page

		AB			
		00	**01**	**11**	**10**
C	**0**	0	0	d	0
	1	1	0	d	0

Three variable Karnaugh map

		AB			
		00	**01**	**11**	**10**
CD	**00**	0	0	0	1
	01	1	1	0	1
	11	d	0	d	0
	10	d	0	d	0

Four variable Karnaugh map

3. Terminology
    a. Literal – each variable in a product term, either uncomplemented or complemented
        i. Example: A in A$\overline{BC}$
    b. Don't cares – combinations of inputs that will never occur (represented by a **d** or **D**)
        i. Thus, the output at that point can be either 1 or 0
        ii. Binary to decimal converters – if they use 4 bits to represent a decimal digit, we'll never see 1010, 1011, 1100, 1101, and 1111 and thus those are don't cares
        iii. Don't cares are useful for simplifying function further
    c. Implicant – product term for which the function is 1 (e.g., 11 for AND)
    d. Prime implicant – the largest possible implicant
        i. Essential prime implicant – prime implicant that contains a 1 that no other prime implicant has
        ii. Don't cares can be included in these
    e. Cover – set of implicants that cover all the 1's in the map
    f. Cost of a circuit – number of gates + the total number of inputs to the gates

4. Minimization
    a. Generate all prime implicants
        i. Draw rectangles around entries that include 1s and not 0s
        ii. Size of rectangles must be powers of 2 (remember, 1 is a power of 2 as well!)
        iii. Make sure rectangles are as large as possible
        iv. Remember that you can wrap around sides
    b. Eliminate prime implicants that overlap until you find the essential implicants
        i. Other considerations: may want to minimize cost

5. Examples
   a. $f_1 = m0 + m1 + m4 + m5 + m7 = \Sigma(0, 1, 4, 5, 7) = \bar{B} + AC$

		AB			
		00	**01**	**11**	**10**
C	**0**	1	0	0	1
	1	1	0	1	1

   b. $f_2 = \Sigma(6, 8, 9, 10, 11, 12, 13, 14) = A\overline{C} + A\overline{B} + BC\overline{D}$

		AB			
		00	**01**	**11**	**10**
CD	**00**	0	0	1	1
	01	0	0	1	1
	11	0	0	0	1
	10	0	1	1	1