Practice Questions for Mar 21, 2019

1. Show the forms of 4×4 ($n = 4$) matrix R for which the code below (discussed in class on Mar 19) returns false for various #(iterations) = 1, 2, ···, 6(= $C(4, 2)$) and also the case where the code returns true after 6 iterations. We have shown the forms for #(iterations) = 1 and 2.

   Use r_{ij} for the value of $R[i][j]$, $i < j$, and use r'_{ij} for $1 - r_{ij}$ for the value of $R[j][i]$ to indicate that $R[i][j] \neq R[j][i]$.

```
for (int i=0; i<R.length; i++)
    for (int j=i+1; j<R.length; j++)
        if (R[i][j] != R[j][i]) return(false);
return(true);
```

$$\begin{bmatrix} ? & r_{11} & ? & ? \\ r'_{11} & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix} \quad \begin{bmatrix} ? & r_{11} & r_{12} & ? \\ r_{11} & ? & ? & ? \\ r'_{12} & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix} \quad \begin{matrix} \cdots \\ \cdots \\ \cdots \\ \cdots \end{matrix} \quad \begin{matrix} \cdots \\ \cdots \\ \cdots \\ \cdots \end{matrix} \quad \begin{matrix} \cdots \\ \cdots \\ \cdots \\ \cdots \end{matrix} \quad \begin{matrix} \cdots \\ \cdots \\ \cdots \\ \cdots \end{matrix}$$

(i) returns false after 1 iteration. (ii) returns false after 2 iterstions. (iii) returns false after 3 iterations. (iv) returns false after 4 iterations. (v) returns false after 5 iterations. (vi) returns false after 6 iterations.

...
...
...
...

(vii) returns true after 6 iterations.

2. Compute the number of matrices of the various forms, and then compute the average #(iterations) = average #(comparisons $R[i][j] \neq R[j][i]$) for the code in Q1 for $n = 4$.

3. Find errors in the argument below, correct the errors, and find the right answer for the sum $S = 1.2^n + 2.2^{n-1} + 3.2^{n-2} + \cdots + n.2^1$. (The 2nd row is obtained from 1st row when divided by 2. The 3rd row is obtained by subtracting 2nd row from 1st row.)

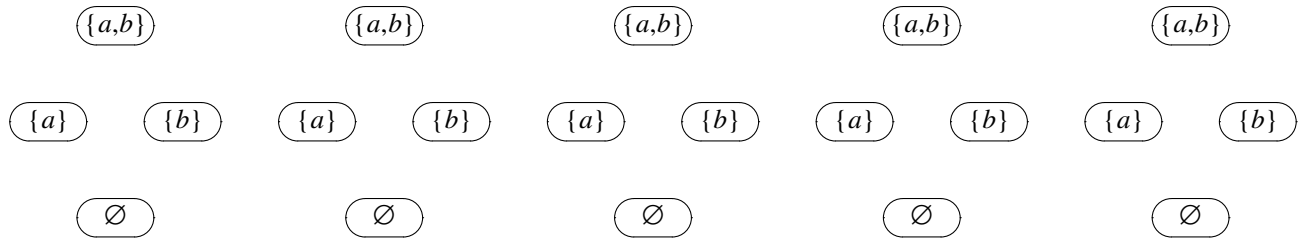
$$\begin{array}{rccccccccccc} S = & 1.2^n & + & 2.2^{n-1} & + & 3.2^{n-2} & + & \cdots & + & n.2^1. & & \\ S/2 = & & & 1.2^{n-1} & + & 2.2^{n-2} & + & \cdots & + & (n\text{-}1)2^1 & + & n.2^0. \\ \hline S/2 = & 2^n & + & 2^{n-1} & + & 2^{n-2} & + & \cdots & + & 2^1 & - & n \\ = & 2(2^{n-1} + 2^{n-2} + \cdots + 1) - n & & & & & & & & & & \\ = & 2(2^n - 1) - n & & & & & & & & & & \end{array}$$

Thus, $S = 4(2^n - 1) - 2n$.

4. Use the result in Q3 to compute the average #(comparisons $R[i][j] \neq R[j][i]$) for the code in Q1 for the general case of $n \times n$ matrices R.

Practice Questions for Mar 19, 2019

1. Consider the "disjoint" relationship between subsets of $S = \{a, b\}$. We write $R_{disjoint}(x, y)$ or $(x, y) \in R_{disjoint}$ when subsets x is disjoint from subset y; we draw all such links (x, y) in the digraph representation for $R_{disjoint}$. Fill-in the links in the diagram in (a) below for the relation $R_{disjoint}$.



(a) $R_{disjoint}$. (b) $R_{disjoint\text{-}or\text{-}equal}$. (c) R_{subset}. (d) $R_{propersubset}$. (e) $R_{subset\text{-}and\text{-}disjoint}$.

Likewise, fill-in the links in the diagram in (b) above for the relation "disjoint or equal", fill-in the links in the diagram in (c) above for the relation "subset", and fill-in the links in the diagram in (d) above for the relation "proper subset", and fill-in the links in the diagram in (e) above for the relation "subset and disjoint".

2. Indicate which of the relations in Q1 that are symmetric. Also, for all non-symmetric relations, mark 'X' every link in the relation's diagram (i.e., digraph) that causes the relation to be non-symmetric.

3. Recall the formula $\sum_d C(n, d)C((n^2 - n)/2, (m - d)/2)$, summed over all $0 \le d \le n$ such that $m - d$ is divisible by 2 and $m - d \le (n^2 - n)/2$, for counting the number of symmetric relations on an n-set having size m, i.e., having m links.

   (a) What does the symbol d represents in the above formulas and why do we have the restriction $0 \le d \le n$?

   (b) Why do we have the factor $C(n, d)$ in the above formula?

   (c) Why do we have the restrictions $m - d$ is divisible by 2 $m - d \le (n^2 - n)/2$? What does $(m - d)/2$ represent?

   (d) What does $(n^2 - n)/2$ correspond to in the above formula.

   (e) Why do we have the factor $C((n^2 - n)/2, (m - d)/2)$ in the above formula.

4. Complement of a relation R on S is a relation R^c on S defined as follow:

   (x, y) is in R^c if and only if (x, y) is not in R.

   Show the diagram of $R_{not\text{-}disjoint} = R^c_{disjoint}$ corresponding to $R_{disjoint}$ in Q1. Argue that R^c is symmetric when R is symmetric. Argue that R^c is non-symmetric when R is non-symmetric. Does this mean that R^c is symmetric if and only if R is symmetric?

5. Based on the answers in Q4, argue that #(symm. rels. of size m on an n-set) = #(symm. rels of size $n^2 - m$ on an n-set) for $(n, m) = (3, 4)$ and $(4, 4)$.

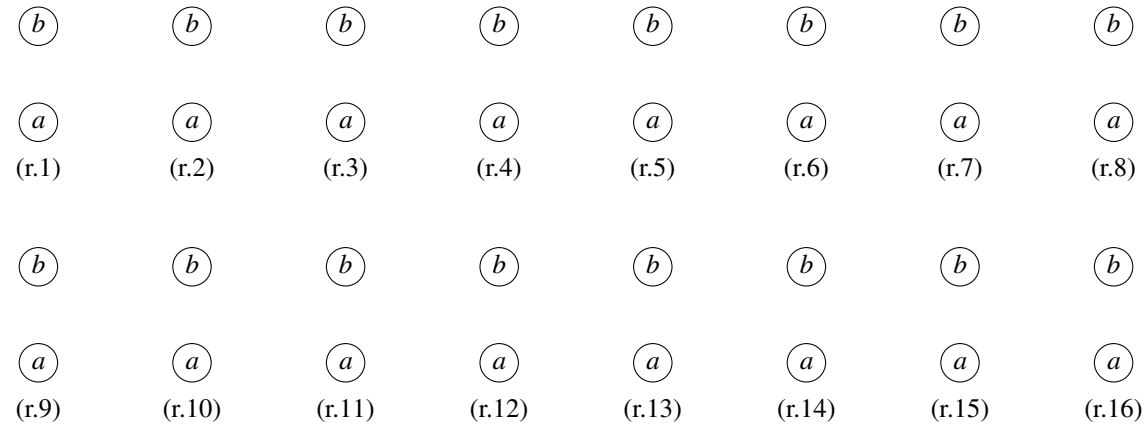
6 Apply the formula in Q3 to determine $S_{n,m}$ = #($n \times n$ symmetric relations with m links) for $n = 3$ and $m = 0, 1, 2, \cdots, 9$. Make a table like the one shown below for computations for $S_{3,m}$'s.

m	d	$C(3, d)$	$C(3, (m - d)/2)$	#(3×3 Symm. rels. with m links)
...	...			

Finally, verify the formula $2^{n(n+1)/2}$ for the total number of symmetric relations on an n-set.

Practice Questions for Mar 14, 2019

1. We have shown there are 2^{n^2} (binary) relations on an n-set. Draw the diagrams for all 16 relations on the 2-set $\{a, b\}$ by filling the links in (r.1)-(r.16) below. As usual, follow some systematic rules to avoid duplicate diagrams; for example, first show all diagrams with no links, then those with just 1 link, then those with just 2 links, and so on.



2. Cross out all relations in Q1 that are not symmetric.
3. Correct all errors in the argument below in counting #(symmetric relations on an n-set).
   (a) For the $n \times n$ matrix $R[i][j]$, $0 \le i,\ j < n$, of a symmetric relation, we can choose the diagonal items and the items above the diagonal arbitrarily (0 or 1).
   (b) The number of diagonal items and the items above the diagnal are $n + (n-1) + (n-2) + \cdots + 2 + 1 = n(n-1)/2$.
   (c) Thus, #(symmetric relations on an n-set) $= 2^{n(n+1)/2} = 2^{n(n+1)/2}$.
4. What does this say about #(non-symmteric relations on an n-set)?
5. Give the number of symmetric relations on a 2-set. How does this match with your answer in Q3?
6. Explain why the statement below

   #(relations on an n-set) > #(symmetric relations on an n-set)

   is not true. Then, give two best possible correct versions of the statement. How will you prove them?
7. Fill in the links below for a largest size (in terms of #(links)) symmetric and a largest non-symmetric relation on $S = \{a, b, c\}$. Do the same for a smallest symmetric and a smallest non-symmetric relation. Which of these are unique? For the non-unique ones, give the number of possible alternatives.

(i) A largest symmetric relation. (ii) A largest non-symmetric relation. (iii) A smallest symmetric relation. (iv) A smallest non-symmetric relation.

8. How many ways can you fill the positions marked '?' in $A = \begin{bmatrix} 1 & 0 & 1 \\ ? & 0 & 0 \\ ? & ? & ? \end{bmatrix}$ and $B = \begin{bmatrix} 1 & 0 & 1 \\ ? & 0 & ? \\ ? & ? & ? \end{bmatrix}$. to get a symmetric relation? How about to get a non-symmetric reltaion?
9. Count the number of relations on an n-set having m links. Verify your answer for $n = 2$ based on your answer in Q1. Repeat the above for symmetric relations and non-symmetric relations.

Practice Questions for Mar 07, 2019

1. Make sure you understand the steps below which show that the average size of subsets of an n-set is $n/2$. (We have shown in the class that this is true for $n = 4$ and 5.)

   (a) #(subsets of size m of an n-set) = $C(n, m)$ for $0 \le m \le n$.

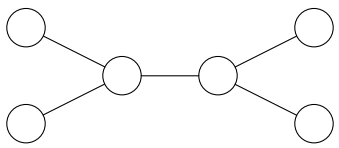
   (b) Total size of all subsets of an n-set

$$
\begin{aligned}
&= 0.\,C(n,0) + 1.\,C(n,1) + 2.\,C(n,2) + 3.\,C(n,3) + 4.\,C(n,4) + \cdots + n.\,C(n,n)\\
&= 1.\,C(n,1) + 2.\,C(n,2) + 3.\,C(n,3) + 4.\,C(n,4) + \cdots + n.\,C(n,n)\\
&= 1.\,n + 2.\,\frac{n(n-1)}{2.1} + 3.\,\frac{n(n-1)(n-2)}{3.2.1} + 4.\,\frac{n(n-1)(n-2)(n-3)}{4.3.2.1} + \cdots + n.\,\frac{n(n-1)(n-2)\cdots 3.2.1}{n(n-1)(n-2)\cdots 3.2.1}\\
&= n.\left[1 + \frac{(n-1)}{1} + \frac{(n-1)(n-2)}{2.1} + \frac{(n-1)(n-2)(n-3)}{3.2.1} + \cdots + \frac{(n-1)(n-2)\cdots 3.2.1}{(n-1)(n-2)\cdots 3.2.1}\right]\\
&= n[C(n-1,0) + C(n-1,1) + C(n-1,2) + C(n-1,3) + \cdots + C(n-1,n-1)]\\
&= n.\,2^{n-1}
\end{aligned}
$$

   (c) This gives the average(size of subsets of an n-set) = $\dfrac{n.\,2^{n-1}}{2^n} = n/2$.

   The equation $C(n, m) = \dfrac{n}{m}\,C(n-1, m-1)$ for $1 \le m \le n$, which is the same as $m.\,C(n, m) = nC(n-1, m-1)$, plays a critial role in showing that the total size of all susset of an n-set is $n.\,2^{n-1}$.

2. Consider the tree below and the labelings of its nodes using the labels $\{a, b, c, d, e, f\}$. As usual each node must have a distinct label. Show two labelings where the two nodes of degree 3 has the labels a and b. How many such labelings are there?



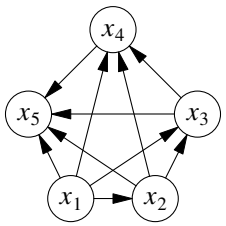
   Now give the total number of labelings of the tree.

   Also, give the total number of labelings of the tree using the labels $\{x_1, x_2, \cdots, x_n\}$, $n \ge 6$?

3. Express the number of items that belong to exactly 1 of A, B, and C in terms of the size of A, B, C, $A \cap B$, $\cdots$, etc. Do the same for the number of items that belong to at least 2 of A, B, and C.

Practice Questions for Feb 21, 2019

1. Consider the following digraph, where we have a link (x_i, x_j) for each $1 \le i < j \le n$ = #(nodes) = 5. We have seen such a digraph before when we let x_i represent the integer i and the links represent "<" relationship between integers. Because there is no cycle in this digraph, such digraphs are called *acyclic* digraph (in short, DAG = directed acyclic graph). Also, because it has the maximum number of links possible for a DAG, it is called a *complete* DAG.



   (a) Give all details of computing the number of (acyclic) x_1x_5-paths.

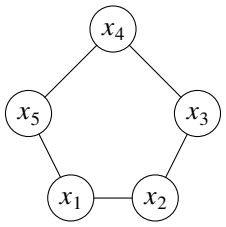
   (b) For general $n \ge 2$, give #(x_1x_n-paths).

2. If G is the complete (undirected) *graph* on n nodes $\{x_i: 1 \le i \le n\}$ and $n \ge 2$, then give a brief argument to show that the number of acyclic x_1x_n-paths is

$$1 + P(n-2, 1) + P(n-2, 2) + \cdots + P(n-2, n-2)$$

   where $P(n, m)$ = #(permutations of m items chosen from a set of n items).

3. Add as many links as possible to the graph below to get a planar graph. The resulting graph is called *maximal* planar graph on 5 nodes (maximal because you cannot add any more links and still keep the graph planar.)



   Give the number of maximal planar graphs on 5 nodes

Practice Questions for Feb 19, 2019

1. We showed that we can form a committee of a president, a secretary, and 3 other members out of 10 people in 10×9×C(8, 3) = 5040 ways. This was achieved by doing the following steps:

   (P) choose the president first (in 10 ways, out of 10 people), then

   (S) choose the secretary (in 9 ways, out of 9 remaining people), and then

   (O) choose the other 3 members (in C(8, 3) = 56 ways, out of remaining 8 people).

   We can call this the ⟨P, S, O⟩ or, in short, PSO approach. One could do the steps P, S, and O in some other order. For example, O first, then S, and then P; We can call this the ⟨O, S, P⟩ or, in short, OSP approach. Show the number of ways to do the step O first, the number of ways to do S next, the number of ways to do P next, and finally the number of ways to form the committee using the OSP approach.

   Repeat the above now for all other approaches.

   There are still other approaches. We can first choose 5 people out of 10 and then use one of PSO, POS, etc approaches applied to those 5 people. In each case, we get the same 5040 number of ways to choose the committes.

2. Consider the code below (it is the same code we considered in the short-quiz on Feb 14).

```
1.  for (int i=0; i<H.length; i++)
2.      if (H[i] > W[i]) return(false);
3.  return(true);
```

   The table below (except the last row) shows the tests done in line 2 for different loop-iterations for H = [1, 1, 0, 1] and the kind of W[] that could make the code exit with the return value **false** in any given iteration. Notes: (1) In col. 5, all previous tests in line 2 must fail (to avoid exiting the loop); in col. 6, the most recent test must succeed to exit the loop. (2) The last row in the table is not an iteration of the loop, and it only has the 2nd column and the last two columns. (3) All W[] types in the last column are disjoint, i.e., no W[] fits more than one type; also, each of 2^4 = 16 W[]'s fits one type.

Iteration	i	Test in line 2 in this iteration	W[i] to satisfy the test and exit loop	All previous tests in line 2 and their successes(T)/failures(F)	What we know about W[] if loop exits now and type of W[]
1	0	H[0] > W[0], i.e., 1 > W[0]	W[0] = 0	None	W[0] = 0; W = [0, ?, ?, ?]
2	1	H[1] > W[1], i.e., 1 > W[1]	W[1] = 0	1 > W[0] (F)	W[0] = 1, W[1] = 0; W = [1, 0, ?, ?]
3	2	H[2] > W[2], i.e., 0 > W[2]	No W[2]; cannot exit	1 > W[0] (F), 1 > W[1] (F)	W[0] = 1, W[1] = 1, W[2] = ?; cannot exit
4	3	H[3] > W[3], i.e., 1 > W[3]	W[3] = 0	1 > W[0] (F), 1 > W[1] (F), 0 > W[2] (F)	W[0] = 1, W[1] = 1, W[2] = ?, W[3] = 0; W = [1, 1, ?, 0]
	4			1 > W[0] (F), 1 > W[1] (F), 0 > W[2] (F), 1 > W[3] (F)	W[0] = 1, W[1] = 1, W[2] = ?, W[3] = 1; W = [1, 1, ?, 1]

3. Repeat Problem 2 for the following H = [0, 1, 0, 0, 1, 1]; also show computation of average #(iterations). Note that #(iterations) for the case of return value **true** is H.length.

4. Repeat Problem 2 for the following code (the loop is now iterated in decreasing values of i) and H = [1, 0, 0, 0, 1, 1]; also show computation of average #(iterations).

```
1.  for (int i=H.length-1; i>=0; i--)
2.      if (H[i] > W[i]) return(false);
3.  return(true);
```

5. Use the formula $C(n, m) = n!/(m!(n-m)!)$ to show $C(m+n+p, m)C(n+p, n) = C(m+n+p, n)C(m+p, m)$. Give two other such equalities.

Practice Questions for Feb 14, 2019

1. Consider the code below. As usual, H and W are two subsets, represented by 0/1-arrays, of a universe of discourse Ω.

```
for (int i=0; i<H.length; i++)
    if (H[i] > W[i]) return(false);
return(true);
```

   Fill the following table to analyze the performance of the above code for the fixed H = [1, 0, 0, 1, 1, 1] and an arbitrary 0/1-array W of length 6. Use '?' to indicate don't care situation in describing the form of W. **Note that there may not be a W for every value (1 to 6) of #(iterations).**

#(iterations)	Return value	Form of W	#(such W's)
1	false	···	···

   Now, based on the table, compute the average #(iterations).

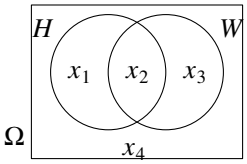
2. What do you think the above code is testing, i.e., when will it return true (express your answer both in English and in set notation)?
3. Repeat Problems 1 and 2 with "H[i] > W[i]" replaced by "H[i] >= W[i]" in the code above.
4. If $A\Delta B = \emptyset$ what does that say about A and B? (Show the possible Venn-diagrams with $A\Delta B = \emptyset$.)

   What can you say for the case $A\Delta B\Delta C = \emptyset$? (Show the possible Venn-diagrams with $A\Delta B\Delta = \emptyset$. Also, give example A, B, C with $A\Delta B\Delta C = \emptyset$; keep them fairly general, i.e., unlike $A = B = C = \emptyset$.)

   What is the relationship between $|A\Delta B|$ and $|A\cup B|$? What is the relationship between $|A\Delta B\Delta C|$ and $|A\cup B\cup C|$?
5. What do we mean by "The symmetric difference operation Δ on sets is an associative operation"? Which of the operations "$\cup$" and "$\cap$" are associative?

Practice Questions for Feb 12, 2019

1. Assume subsetOne = [1, 1, 0, 0, 1] and we are testing whether subsetTwo (which is also a 0/1-array of length 5 = subsetOne.length) equals subsetOne or not.
   (a) Give the number of 0/1-arrays subsetTwo for which the first iteration of the loop (see the code given in an earlier practice question) returns the value false.
   (b) Repeat (a) for $k = 2, 3, \cdots, 5$.
   (c) Compute the total number of loop-iterations for returning the value false for different subsetTwo.
   (d) Compute the average number of loop-iterations for returning the value false for different subsetTwo.
   (e) What is #(loop-iterations for the return value true)? What is #(0/1-arrays subsetTwo that give return value true)?
   (f) Compute the average number of loop-iterations for the code (irrespective of the return value)
2. Give the average values in Problem 1 for the general case (subsetOne is arbitrary 0/1-array of length $n \geq 1$)?
3. Consider two subsets H (for things John has) and W (for things that John wants) of $\Omega \neq \emptyset$, and the four disjoints subsets $H - W$, $H \cap W$, $W - H$, and $H^c \cap W^c$ into which Ω is partitioned (see the Venn-diagram below).



   We write x_1 = "yes", "no", and "?" to indicate, respectively, that $H - W \neq \emptyset$, $= \emptyset$, and we do not know (care) whether $H - W$ is empty or not. Similarly for other x_i's. Note that $\Omega \neq \emptyset$ implies we cannot have $x_1 = x_2 = x_3 = x_4$ = "no".

   Shown below are a few cases of yes/no/? combinations of values of x_i's, the description of the situation in ordinary English, and the description in the set notation. (A few cases are completely filled and a few are only partially filled.) In some situations, you may need a long sentence combining several other cases. Note that #9 is a special case of #1, and we need a longer English expression to describe this situation than that for #1; the set notation also shows that #9 is a special case of #1. Similar remarks hold for case #10.

   Give English sentence and set-expression for each of #3, #5, #6, and #7. Do the same for some of the cases with two yes/no's and two "?". How many (total) different cases are there with all yes/no/? combinations (except all no's)? What is English descriptiion for the case "? ? ? ?" and what is the set-expression for this case?

Case	x_1	x_2	x_3	x_4	English description (without using set terminology)	Set notation (without use of complement)
1.	yes	?	?	?	John has some thing that he does not want.	$H \not\subseteq W$
2.	no	?	?	?	John doesn't have any thing that he doesn't want, i.e., John has only things that he wants. (John may not want any thing or have any thing.) Another way of expressing it without using negation: John wants everything he has.	$H \subseteq W$
3.	?	yes	?	?		
4.	?	no	?	?	John does not have any thing he wants. Another way of saying it: John does not want any thing he has.	$H \cap W = \emptyset$
5.	?	?	yes	?		
6.	?	?	no	?		
7.	?	?	?	yes		
8.	?	?	?	no	There is nothing that John does not have and does not want.	$H \cup W = \Omega$
9.	yes	yes	?	?	··· ···	$H \not\subseteq W$ and $H \cap W \neq \emptyset$
10	yes	no	?	?		
...	...	...	...			

4. Draw Venn-diagrams to show that $(A \Delta B) \Delta C = A \Delta (B \Delta C)$. This is why we can write $A \Delta B \Delta C$ to denote either of these. Describe in English the set $A \Delta B \Delta C$ in a clean compact form.

Practice Questions for Feb 05, 2019

1. What property between subsetOne and subsetTwo will make the following code return true? Illustrate with an example of subsetOne and subsetTwo; also, give #(comparison "subsetOne[i] == subsetTwo[i]" is done). If the return-value is false, then what can you say about #(comparison "subsetOne[i] == subsetTwo[i]" is done)?

```
for (int i=0; i<subsetOne.length; i++)
    if (subsetOne[i] == subsetTwo[i]) return (false);
return (true);
```

2. Give a code to compute the size of subsetOne $\cap$ subsetTwo. How many times will the loop iterate?

Practice Questions for Jan 31, 2019

1. To compute $C(n, m)$ using the formula $C(n, m) = \frac{n}{m} C(n-1, m-1)$, assuming $m \le n/2$, we succesively compute $N_1 = \frac{n-m+1}{1} = n-m+1$, $N_2 = (n-m+2) * N_1/2$, $N_3 = (n-m+3)N_2/3$, and so on, finally $N_m = n * N_{m-1}/m = C(n, m)$.
   (a) Give a for-loop to carry out this computation of $C(n, m)$; use the integer-variable name "result" for the successive values $N_1, N_2, \cdots, N_m$ computed.
   (b) Express the number of multiplications and divisions done in terms of n and m.
   (c) Repeat (a)-(b) for computing $C(n, m)$ based on the formula $C(n, m) = \frac{n-m+1}{m} C(n, m-1)$.
   (e) Is there any advantage of the for-loop in (a) vs. that in (c) in terms of #(iterations), #(multiplication and division operations), and the nature of intermediate values (being large vs. small) of result? Justify your answer with examples.

2. We know a subset can be represented by an array of 1's and 0's (1 for an item in the subset and 0 for an item not in the subset). Shown below are some example subsets of {A, B, C, D, E} and their array-representation. Note that #(1's in the array) equals the size of the subset.

Subsets	A	B	C	D	E	Array representation
$\{A, B, E\}$	1	1	0	0	1	[1, 1, 0, 0, 1]
$\{B\}$	0	1	0	0	0	[0, 1, 0, 0, 0]
$\varnothing$	0	0	0	0	0	[0, 0, 0, 0, 0]

   What does the following code return when subsetOne and subsetTwo equal the arrays for subsets $\{A, B, E\}$ and $\{B\}$?

```
for (int i=0; i<subsetOne.length; i++)
    if (subsetOne[i] != subsetTwo[i]) return (false);
return (true);
```

   How many times the comparison "subsetOne[i] != subsetTwo[i]" is done?

   State in English what the code returns, in general, for two input arrays subsetOne and subsetTwo (of 0's and 1's and having the same length)?

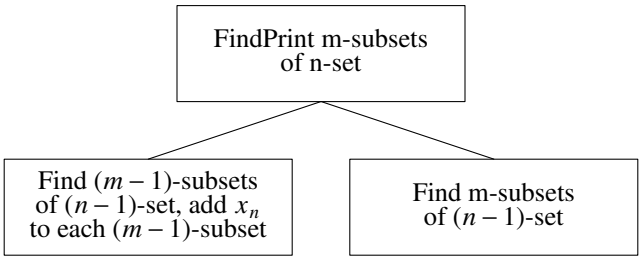
   For subsetOne = [1, 1, 0, 0, 1], how many subsetTwo give the return value "true" and what is #(comparison "subsetOne[i] != subsetTwo[i]") for each of those subsetTwo?

   For subsetOne = [1, 1, 0, 0, 1], how many subsetTwo give the return value "false" and how many of them have #(comparison "subsetOne[i] != subsetTwo[i]") = k for each of k = 1, 2, 3, and 4?

   Give the codes to test each of the following: (a) subsetOne is a subset of subsetTwo, and (b) subsetOne and subsetTwo are disjoint.

3. We know that 4 straightlines L1, L2, L3, and L4 can intersect in minimum 0 points and maximum 6 = C(4,2) points. Show whether it is possible for 4 straightlines to intersect in exactly k points for k = 1, 2, 3, 4, and 5. (Note that n straightlines L1, L2, ..., Ln can intersect in maximum C(n,2) = n(n-1)/2 points.)

Use $C(n, m) = C(n-1, m-1) + C(n-1, m)$ for $m \geq 0$
and otherwise, return the empty-subset



Alternate approach:

1. Find the first set.
2. Find repeatedly the next subset until there is no more.

Finding 3-subsets of 5-set:

$$
\begin{aligned}
00111 &= \{x_3, x_4, x_5\} \\
01011 &= \{x_2, x_4, x_5\} \\
01101 &= \{x_2, x_3, x_5\} \\
01110 &= \{x_2, x_3, x_4\} \\
10011 &= \{x_1, x_4, x_5\} \\
10101 &= \{x_1, x_3, x_5\} \\
10110 &= \{x_1, x_3, x_4\} \\
11001 &= \{x_1, x_2, x_5\} \\
11010 &= \{x_1, x_2, x_4\} \\
11100 &= \{x_1, x_2, x_3\}
\end{aligned}
$$

Parallel Computing.

1. Consider computing all $c(n, m)$ for $0 \le m \le n \le N$ for a fixed $N = 7$, say, using the first method (two loops to intialize values 1 and another nested-loop for assigning other values). If we have 2 CPU's to do the computations, then we could do the work as follow. For simplicity, assume that each assignment, reading value of an $C(n, m)$, and adding two values like $C(n-1, m-1)$ and $C(n-1, m)$ take 1 unit of time. We ignore other coputation time for testing things like "$i \le N$" and updating 'i++", etc.

   For initializing C[i][0]'s:

Time	1	2	3	4
CPU#1	C[0][0] = 1	C[2][0] = 1	C[4][0] = 1	C[6][0] = 1
CPU#2	C[1][0] = 1	C[3][0] = 1	C[5][0] = 1	C[7][0] = 1

   For initializing C[i][i]'s:

Time	5	6	7	8
CPU#1	C[1][1] = 1	C[3][3] = 1	C[5][5] = 1	C[7][7] = 1
CPU#2	C[2][2] = 1	C[4][4] = 1	C[6][6] = 1	

   For the nested-loop, doing the operation of j-loop in parallel for each i

i = 2, Time	5-7
CPU#1	readind C[1][0] and C[1][1], adding them, and assigning to C[2][1]
CPU#2	

i = 3, Time	8-10
CPU#1	readind C[2][0] and C[2][1], adding them, and assigning to C[3][1]
CPU#2	readind C[2][1] and C[2][2], adding them, and assiging to C[3][2]

i = 4, Time	11-13	14-16
CPU#1	readind C[3][0] and C[3][1], adding them, and assigning to C[4][1]	readind C[3][2] and C[3][3], adding them, and assigning to C[4][3]
CPU#2	readind C[3][1] and C[3][2], adding them, and assiging to C[4][2]	

   (a) Show the parallel operation-timing for i = 5, 6, and 7 for the nested j-loop.

   (b) Show the total amount of time taken by 1 CPU and that by 2 CPU's.

   (c) IS it true that the time for 2 CPU's is about 1/2 the time for 1 CPU?

   (d) If N is large and we have 3 CPU's, do you think the total time would be about 1/3 of the time for 1 CPU?

Practice Questions for Jan 29, 2019

1. We had two loops shown below on left to set the values $C[n, 0] = C[n, n] = 1$ for $0 \le n \le N$. What is the advantage (if any, other than the code being shorter) of replacing the two loops by one loop as shown on right?

```
for (int i=0; i<=N; i++)          C[0][0] = 1;
    C[i][0] = 1;                  for (int i=1; i<=N; i++)
for (int i=1; i<=N; i++)              C[i][0] = C[i][i] = 1;
    C[i][i] = 1;
```

   (We replaced $C[i, j]$ here by $C[i][j]$ because the latter is the proper way of writing items of a 2-dimensional array or matrix. Also, as a convention, we should start variable names by a lower-case letter, but we will continue to use $C[i][j]$ instead of, say, $c[i][j]$. Likewise, we continue to use N.)

2. We used the following loop to assign values to the remaining $C[i][j]$'s. Rewrite the loop to take advantage of the symmetry-property $C(n, m) = C(n, n-m)$ of $C(n, m)$. (This will reduce the number of addition operations involving the numbers $C[i][j]$ by a factor of 2 approximately.)

```
for (int i=2; i<=N; i++)
    for (int j=1; j<i; j++)
        C[i][j] = C[i-1][j-1] + C[i-1][j];
```

3. Complete the code below for a function `int combination(int n, int m)` for inputs $0 \le m \le n$ to return the value of $C(n, m)$.

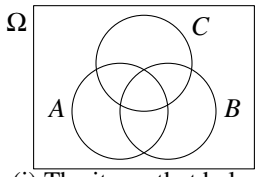
```
public static int combination(int n, int m) //assume 0 <= m <= n
{ ...
  ...
  ...
}
```

Practice Questions for Jan 24, 2019

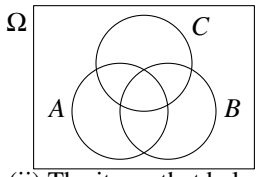
1. Consider three subsets A, B, and C or Ω (see the previous practice questions). Answer the following counting questions in terms $|A|$, $|B|$, $|C|$, $|A \cap B|$, $|A \cap C|$, $|B \cap C|$, $|A \cap B \cap C|$, and $|\Omega|$, as needed. Give the simplest (most simplified) formula possible.
   (a) Show the size of the set of items in A that are not in B or C.
   (b) Show the size of the set of items that are in exactly one of A or B or C.
   (c) Show the size of the set of items in both A and B but not in C.
   (d) Show the size of the set of items that are exactly in two of A, B, and C.
2. Use the formulas $mC(n, m) = (n - m + 1)C(n, m-1)$ and $C(n, m) = C(n-1, m-1) + C(n-1, m)$ to obtain the formula $C(n, m) = \dfrac{n}{m} C(n-1, m-1)$ for $1 \le m \le n$.
3. Consider an n-set $S_n = \{x_1, x_2, \cdots, x_n\}$ and, likewise, an $(n-1)$-set S_{n-1}. Draw the lines connecting all m-subsets and $(m-1)$-subsets of S_{n-1} to m-subsets of S_n for $n = 5$ and $m = 2$. (Write the subsets in systematic order.}
4. Draw the lines connecting all m-subsets of S_n to $m-1$-subsets of S_n for $n = 5$ and $m = 2$. (Write the subsets in systematic order.}

Practice Questions for Jan 22, 2019

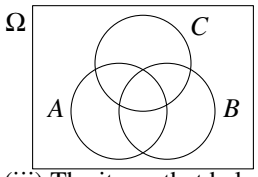
1. Shade the appropriate part of each Venn-diagram below based on its caption. For each case, show the formula to compute the corresponding cardinality in terms of $|A|$, $|B|$, $|C|$, $|A \cap B|$, $|A \cap B \cap C|$, etc. Here, Ω = Universe of Discourse (see the text-book for Universe of Discourse, if needed).



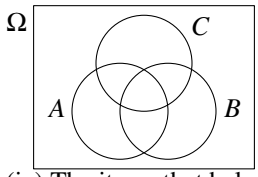
(i) The items that belong to ≥ 1 of A, B, and C.



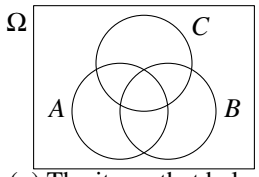
(ii) The items that belong to ≥ 2 of A, B, and C.



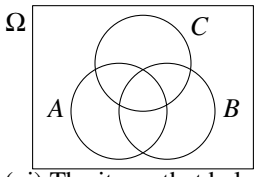
(iii) The items that belong to exactly 1 of A, B, and C.



(iv) The items that belong to ≤ 1 of A, B, and C.



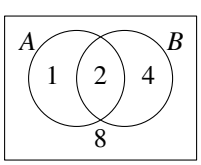
(v) The items that belong to ≤ 2 of A, B, and C.



(vi) The items that belong to exactly 2 of A, B, and C.

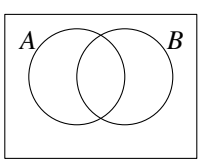
2. The Venn-diagram below shows two subsets A, $B \subseteq \Omega$ = Universe of Discourse. The numbers in the diagram show the size of the various subsets; thus, $|A - B| = 1$, $|A \cap B| = 2$, $|B - A| = 4$, and $|\Omega - (A \cup B)| = 8$. We chose these sizes so that the sizes of $16 = 2^{2^2}$ subsets X_1 to X_{16} of Ω that can be formed from A and B using union, intersection, and complement are distinct. We write the intersection $X \cap Y$ below simply as XY; thus, $AA^c = A \cap A^c = \varnothing = BB^c$.

The sizes of various subsets $A - B$, etc.

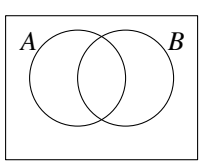


Shade the part of the venn-diagram in each of 16 cases below (continued to next page) representing the given set X_i and show the size $|X_i|$.

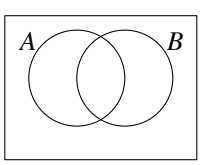
$X_1 = AA^c$
$|X_1| =$



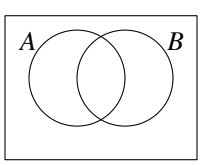
$X_2 = A$
$|X_2| =$



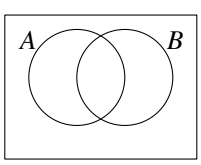
$X_9 = AB$
$|X_9| =$



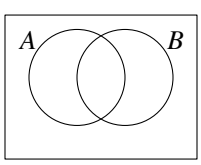
$X_{10} = A^c B$
$|X_{10}| =$



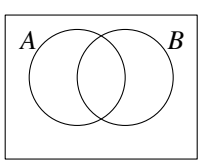
$X_{11} = A \cup B^c$
$|X_{11}| =$



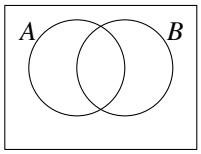
$X_{12} = A^c B^c$
$|X_{12}| =$



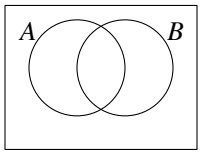
$X_{13} = A^c \cup B$
$|X_{13}| =$



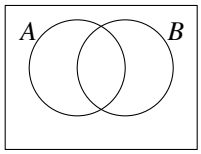
$X_{14} = AB \cup A^c B^c$
$|X_{14}| =$



$X_{15} = AB^c \cup A^c B$
$|X_{15}| =$

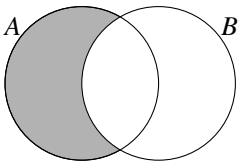


$X_{16} = A \cup A^c$
$|X_{16}| =$

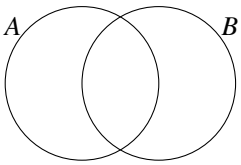


Practice Questions for Jan 17, 2019

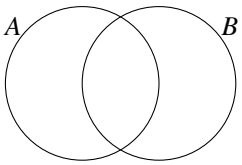
1. In part (i) below, we have shown two sets A and B, and the area for the set $A - B$ is shaded. Shade the areas for the sets indicated in (ii)-(iv).



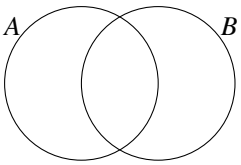
(i) The shaded area is for the set $A - B$



(ii) Shade the area for the set $B - A$

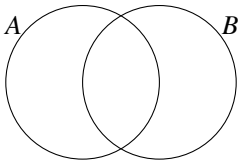


(iii) Shade the area for the set $A \cup B$



(iv) Shade the area for the set $A \cap B$

2. Shade the area below for elements (items) that belong to exactly one of the sets A and B. The set of such elements is denoted by $A \Delta B$ and is called the *symmetric difference* of A and B.



3. If $|A| = 10$, $|B| = 8$, and $|A \cap B| = 3$, then give the values of the following:
   (a) $|A - B|$ = ...
   (b) $|B - A|$ = ...
   (c) $|A \Delta B|$ = ...

4. Complete the equations below for arbitrary sets A and B in terms of $|A|$ and $|B|$ (similar to the ones we did in the class for $|A \cup B|$).
   (a) max $|A \cap B|$ = ...
   (b) min $|A \cap B|$ = ...
   (c) $|A \Delta B| = |A| + |B| -$ .....

5. Which of the following are true?
   (a) $A \Delta B = (A - B) \cup (B - A)$
   (b) $A = B$ if and only if $A \Delta B = \varnothing$.
   (c) $A - B = A - (A \cap B)$.
   (d) The sets $A - B$ and $B - A$ are disjoint.
   (e) $A = (A - B) \cup (A \cap B)$.
   (f) $|A| = |A - B| + |A \cap B|$.

6. Is the following proof of $|A \cup B| = |A| + |B| - |A \cap B|$ valid (correct)? If not, explain why not.

   Step 1. It is clear that if A and B are disjoint sets, then $|A \cup B| = |A| + |B|$.
   Step 2. Because $A \cup B = A \cup (B - A)$, we have $|A \cup B| = |A \cup (B - A)|$.
   Step 3. Because the sets A and $B - A$ are disjoint, from (1) we get $|A \cup (B - A)| = |A| + |B - A|$.
   Step 4. Because $B = (B - A) \cup (A \cap B)$, we get $|B| = |(B - A) \cup (A \cap B)|$.
   Step 5. Because the sets $B - A$ and $A \cap B$ are disjoint, from (1) we get $|(B - A) \cup (A \cap B)| = |B - A| + |A \cap B|$.
   Step 6. Combining the results in steps (2)-(3), we get $|A \cup B| = |A| + |B - A|$.
   Step 7. Combining the results in steps (4)-(5), we get $|B| = |B - A| + |A \cap B|$, i.e., $|B - A| = |B| - |A \cap B|$.
   Step 8. Combining the results in steps (6)-(7), we get $|A \cup B| = |A| + |B| - |A \cap B|$.

PRACTICE QUESTIONS (on the importance of counting various things; bring written solutions to the class on Jan 15):

1. Give 3 good reasons for determining the number of students in a course (i.e., 3 good uses of this number).
2. Give 3 good reasons for determining the number of students attending LSU in Spring 2019.
3. Give 3 things that we should count in a computer program to get an idea about it.
4. Show all possible ways you can choose a manager and an assistant manager out of 4 candidates Adam, Betty, Cindy, and David.