CS 173: Discrete Structures, Spring 2014
Final review solutions

These problems should not be turned in. They are to help you review for the final.

1. **Counting problems**

   (a) Let $M = \{a, b, c, d\}$. How many different partitions of M are there?
   Solution: There are 15 different ways to partition M:

   - All the elements together $\{\{a, b, c, d\}\}$ (1 choice)
   - Three elements together and one separate, e.g. $\{\{a\}, \{b, c, d\}\}$. (4 choices)
   - Two pairs of elements e.g. $\{\{a, b\}, \{c, d\}\}$. (3 choices)
   - A pair of elements and two separate elements e.g. $\{\{a\}, \{b\}, \{c, d\}\}$. (6 choices)
   - All four elements separate $\{\{a\}, \{b\}, \{c\}, \{d\}\}$ (1 choice)

   (b) In the game Tic-tac-toe (played on the usual 3 by 3 grid), how many different board configurations are possible after four moves (i.e. two moves by each player)?
   Solution: There are $\binom{9}{2}$ ways to pick a set of positions to contain the X's. Given that choice, there are $\binom{7}{2}$ ways to pick two more positions to contain the O's. So the total number of configurations is $\binom{9}{2}\binom{7}{2} = \frac{9!}{2!2!5!}$. Or you can think of this as the number of ways to rearrange 2 X's 2 O's and 5 blanks in 9 spots so $\binom{9}{2\ ,\ 2} = \binom{9}{2\ ,\ 5}$.

   (c) Suppose a car dealer is planning to buy a set of Civics, Accords, and Fits (three kinds of cars). The dealer will buy ten cars in total and can buy any number of each type. How many different choices does he have? The sets are unordered, so three Civics and seven Fits is the same as seven Fits and three Civics.
   Solution: He has $\binom{(10+2)}{2}$, i.e. $\binom{12}{2}$ ways to choose his set of cars.

   (d) Suppose a set S has 11 elements. How many subsets of S have an even number of elements? Express your answer as a summation. (You do not need to simplify expressions involving permutations and/or combinations.)
   Solution: Subsets of S having an even number of elements would have $0, 2, 4, 6, 8$ or 10 elements. There are $C(11, 0)$ subset of S with no elements, $C(11, 2)$ subsets of S with 2 elements, and so on. So the number of subsets of S having an even number of elements is

   $$\binom{11}{0} + \binom{11}{2} + \binom{11}{4} + \binom{11}{6} + \binom{11}{8} + \binom{11}{10} = \sum_{i=0}^{5} \binom{11}{2i}.$$

   (e) If $x, y, z \in \mathbb{N}$, how many solutions are there to the equation $x + y + z = 25$?
   Solution: Imagine that the number 25 represents 25 objects that can be chosen in three different types: type x, type y, and type z. To indicate type we divide the objects into three bins, i.e., place 3 dividers into the list of objects. The number of

ways to place the objects into the bins is the number of different solutions and is given by the expression for combinations with repetition

$$\binom{25+3-1}{25} = \binom{27}{25} = \binom{27}{2} = 351$$

(f) Suppose that A is a set containing p elements and B is a set containing n elements. How many functions are there from A to $\mathbb{P}(B)$? How many of these functions are one-to-one?

Solution: $\mathbb{P}(B)$ contains 2^n elements. So the total number of functions from A to $\mathbb{P}(B)$ is $(2^n)^p$. The number of one-to-one functions is $\frac{2^n!}{(2^n-p)!}$.

2. **Big O**

Sort the following 7 functions of n by asymptotic complexity. That is, write f to the left of g if and only if $f(n) \ll g(n)$.

$$3n\log(n^3) \qquad 3^n+n \qquad 17^n+25 \qquad n^2 \qquad 20n^2\log n \qquad 173 \qquad 7n!+2$$

Solution:
$173 \ll 3n\log(n^3) \ll n^2 \ll 20n^2\log n \ll 3^n+n \ll 17^n+25 \ll 7n!+2$

3. **Binomial theorem**

(a) How many terms are contained in $(x+y+z)^{30}$ after carrying out all multiplications, but before collecting like terms?

Solution: Note that $(x+y+z)^0 = 1$ which has $1 = 3^0$ terms. Similarly, $(x+y+z)^1 = x+y+z$ which has $3 = 3^1$ terms. Each time we increase the exponent by one, we are multiplying each of the terms from the previous expression by each of x, y, and z tripling the total number of terms. Thus, $(x+y+z)^{30}$ has 3^{30} terms.

(b) How many terms are contained in $(x+y+z)^{30}$ after carrying out all multiplications and collecting like terms?

Solution: After performing the multiplications and collecting similar terms, we are left with a sum of terms of the form $C \cdot x^i y^j z^k$, where $C, i, j, k \in \mathbb{N}$ and $i+j+k = 30$. Think of each term as having exactly 30 slots for variables (all to the first power) where we can choose from a list of three variables. Then the exponents i, j, k tell you the number of copies of x, y, z (respectively) that show up in each term. The number of ways to choose a string of length 30 from the alphabet list $\{x, y, z\}$ with repetition where order does not matter (since multiplication is commutative) is

$$\binom{30+3-1}{30} = \binom{32}{30} = \frac{32!}{30! \cdot (32-20)!} = \frac{32 \cdot 31}{2} = 496.$$

(c) What is the coefficient of the $x^{15}y^6z^9$ term? (You may leave you answer in terms of factorials!)

Solution: The coefficient is the number of permutations with repetition of a list of elements containing 15 x's, 6 y's, and 9 z's. This number is given by

$$\binom{30}{15\ ,\ 6} = \binom{30}{15} \cdot \binom{15}{6} \cdot \binom{9}{9} = \frac{30!}{15! \cdot 6! \cdot 9!}.$$

4. **Algorithms**

Consider the following procedure Gnarly, which returns true or false. Notice the indenting on lines 04-05: they execute only when the test on line 03 succeeds. You can assume that $n \geq 1$ and that extracting a subarray (e.g. in the recursive calls in line 06) requires only constant time.

```
01 Gnarly (a_1, ..., a_n: array of integers)
02      if (n = 1) return true
03      else if (n = 2)
04            if (a_1 = a_2) return true
05            else return false
06      else if (Gnarly(a_1, ..., a_{n-1}) and Gnarly(a_2, ..., a_n)) return true
07      else return false
```

(a) If Gnarly returns true, what must be true of the values in the input array? Briefly justify your answer.

Solution: All the values in the input array must be equal. The base cases (lines 02-05) check that that is true for very small arrays. If the array has more than three elements, then verifying that all the elements are equal when you remove each of two different elements (line 06) forces all the elements to be equal in the whole array.

(b) Give a recursive definition for $T(n)$, the running time of Gnarly on an input of length n. Be sure to include a base case.

Solution:

$$\begin{aligned} T(1) &= T(2) = c \\ T(n) &= 2T(n-1) + d \end{aligned}$$

(c) Give a big-theta bound on the running time of Gnarly. For full credit, you must show work or briefly explain your answer.

Solution: Gnarly takes $\Theta(2^n)$ time. This recurrence is basically the same as that for the Towers of Hanoi solver. Each node in the recursion tree contains only constant work, but there are $2^{n+1} - 1$ nodes in the tree.

5. **Algorithms**

Here is pseudocode for functions `find_closest` and `find_closest_sorted`. Each function takes as input an array of n numbers and a target number and outputs the closest value in the input that is at least as large as the target (or NULL if no such number exists). `find_closest_sorted` assumes that the input array is sorted in ascending order.

```
01 function find_closest(input_array, target)
02 sorted_array = cocktailsort(input_array);   // cocktailsort takes Θ(n^2) time
03 return find_closest_sorted(sorted_array, target);

04 function find_closest_sorted(sorted_array, target)
05 length = get_length(sorted_array) // takes constant time
06 if length = 0
07     return NULL;
08 else if length = 1
09     return sorted_array[0];
10 mid = ⌊length/2)⌋;
11 if sorted_array[mid-1]<target
12     half_array = sorted_array(mid ... length-1);  // array containing second half of elements
13 else
14     half_array = sorted_array(0 ... mid-1);   // array containing first half of elements
15 return find_closest_sorted(half_array, target);
```

Answer the questions below using worst-case analysis (e.g., the target value is equal to the largest number in the input array). Assume that the input length n is a power of 2.

(a) Suppose $R(n)$ is the running time of `find_closest_sorted`. Give a recursive definition of $R(n)$. Assume that it takes constant time to extract the first or second half of the array in lines `12` and `14`.

Solution:

$$R(0) = R(1) = d$$
$$R(n) = R(n/2) + c$$

(b) What is the height of the recursion tree for $R(n)$?

Solution: $\log n$

(c) How many leaves are in the recursion tree for $R(n)$?

Solution: 1

(d) What is the big-Theta (aka tight big-O) running time of `find_closest_sorted`? Make your answer as simple as posible.

Solution: $\Theta(\log n)$

(e) What is the big-Theta running time of `find_closest`. Make your answer as simple as possible.

Solution: $\Theta(n^2)$

6. **Power sets**

Suppose you were given the following sets:

$$\begin{aligned}\mathbf{A} &= \{\text{Vine}, \text{Tree}, \text{Shrub}\}\\ \mathbf{B} &= \{\{\text{Tree}\}\}\\ \mathbf{C} &= \{\text{Vine}, \text{Moss}\}\\ \mathbf{D} &= \{\text{Red}, \text{Green}\}\\ \mathbf{E} &= \{\text{Red}\}\end{aligned}$$

For the following expressions, list the elements of the set or give the cardinality (as appropriate):

(a) $D \times C$

(b) $\mathbb{P}(B \cup E)$

(c) $|A \times \ \mathbb{P}(A \cup D)\ |$

(d) $\{S \in \mathbb{P}(A \cup D) \ : \ |S| \text{ is a multiple of } 4\}$

(e) $\mathbb{P}(\mathbb{P}(\mathbb{P}(\emptyset)))$

Solution:

(a) $D \times C = \{(\text{Red}, \text{Vine}), (\text{Red}, \text{Moss}), (\text{Green}, \text{Vine}), (\text{Green}, \text{Moss})\}$

(b) $\mathbb{P}(B \cup E) = \{\emptyset, \{\text{Red}\}, \{\{\text{Tree}\}\}, \{\{\text{Tree}\}, \text{Red}\}\}$

(c) $|A \times \ \mathbb{P}(A \cup D)\ | = |A| \cdot |\mathbb{P}(A \cup D)| = 3 \cdot 2^5 = 96$

(d) $\{S \in \mathbb{P}(A \cup D) \ : \ |S| \text{ is a multiple of } 4\} = \{\emptyset, \{\text{Vine}, \text{Tree}, \text{Shrub}, \text{Red}\},$
$\{\text{Vine}, \text{Tree}, \text{Shrub}, \text{Green}\}, \{\text{Tree}, \text{Shrub}, \text{Green}, \text{Red}\},$
$\{\text{Vine}, \text{Shrub}, \text{Green}, \text{Red}\}, \{\text{Vine}, \text{Tree}, \text{Green}, \text{Red}\}\}$

(e) $\mathbb{P}(\mathbb{P}(\mathbb{P}(\emptyset))) = \mathbb{P}(\mathbb{P}(\{\emptyset\})) = \mathbb{P}(\{\emptyset, \{\emptyset\}\}) = \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \{\emptyset, \{\emptyset\}\}\}$

7. **Set-valued functions**

Let $f : X \to Y$ be any function, and let A and B be subsets of X. For any subset S of X define its image $f(S)$ by $f(S) = \{f(s) \in Y \mid s \in S\}$.

(a) Prove that $f(A \cap B) \subseteq f(A) \cap f(B)$. You **must** use the technique of choosing an element from the smaller set and showing that it is also a member of the larger set.

Solution: Note that $f(A \cap B)$ and $f(A) \cap f(B)$ are *sets*. Suppose y is an arbitrary element of $f(A \cap B)$. By the definition of the image of a set, there is an element $x \in A \cap B$ such that $f(x) = y$. Since $x \in A \cap B$, we know $x \in A$ and $x \in B$. Thus, $y \in f(A)$ and $y \in f(B)$. So $y \in f(A) \cap f(B)$. Since y was arbitrary, we conclude that $f(A \cap B) \subseteq f(A) \cap f(B)$.

(b) Prove that it's not necessarily the case that $f(A) \cap f(B) \subseteq f(A \cap B)$ by giving specific **finite** sets and a specific function for which this inclusion does not hold.

Solution: Let $X = \{a, b\}$ and $Y = \{c\}$. Define $f : X \to Y$ by $f(x) = c$ for all $x \in X$. Choose $A = \{a\}$, and $B = \{b\}$. Then $A \cap B = \emptyset$, so $f(A \cap B) = \emptyset$. However, $f(A) = Y = f(B)$, so $f(A) \cap f(B) = Y$.

8. **Proof by contradiction**

(a) State the negation of the following claim. Your answer should be in words, with all negations (e.g. "not") on individual predicates.

For any berry b, if birds like b, then b is edible or b is red.

Solution: There is a berry b such that birds like b but b isn't edible and b isn't red.

(b) Use proof by contradiction to show that $\sqrt{2} + \sqrt{3} \leq 4$.

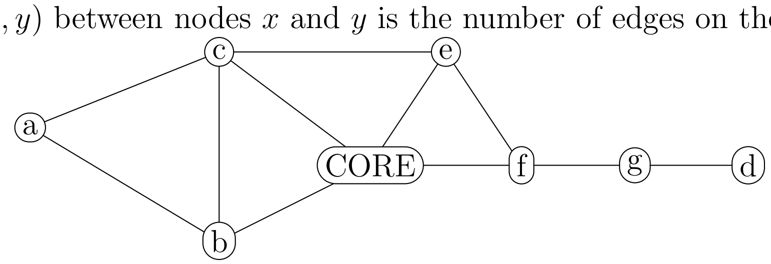
Solution: Suppose not. That is, suppose that $\sqrt{2} + \sqrt{3} > 4$.

Squaring both sides, we get that $2 + 2\sqrt{6} + 3 > 16$. So $2\sqrt{6} > 11$. Squaring again, we find that 24 ¿ 121. This is clearly impossible.

Since assuming $\sqrt{2} + \sqrt{3} > 4$ led to a contradiction, it must be the case that $\sqrt{2} + \sqrt{3} \leq 4$.

9. **Partitions**

Graph G is shown below with set of nodes V is shown below. Recall that the distance $d(x, y)$ between nodes x and y is the number of edges on the shortest path from x to y.



Now let's define:

$$\begin{aligned} M &= \{1, 2, 3, 4\} \\ f(k) &= \{n \in V \ : \ d(n, CORE) = k\} \\ P &= \{f(k) \mid k \in M\} \end{aligned}$$

(a) Fill in the following values:

f(1) =

Solution: $f(1) = \{b, c, e, f\}$

f(3) =

Solution: $f(3) = \{d\}$

(b) Is P a partition of V? For each of the three conditions required to be a partition, explain why P does or doesn't satisfy that condition.

Solution: Notice that

$P = \{f(1), f(2), f(3), f(4)\} = \{\{b, c, e, f\}, \{a, g\}, \{d\}, \emptyset\}$

P is not a partition of V. There is no partial overlap among the elements of P. However, P contains the empty set. Also, the elements of P don't cover all the members: the node CORE is missing.

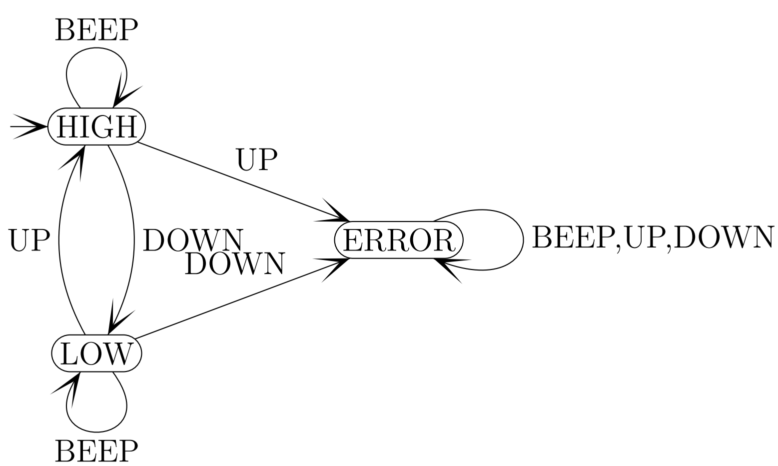
10. **State Diagrams**

(a) Suppose we have a state diagram with n states and k different actions. In how many different ways could we construct a transition function for this diagram?

Solution: The domain of the transition function contains all the pairs of a state plus an action, so it has size nk. The co-domain of the function contains all sets of states, so it has size 2^n. We can choose the output value independently for each input value, so we have $(2^n)^{nk}$ total choices for constructing the transition function.

(b) Suppose we're modelling an RC crane which is receiving a sequence of input commands, each of which is UP, DOWN, or BEEP. This crane has only two vertical positions, and starts in the high position. It should go into an ERROR end state if it is asked to go UP when it is in the high position or DOWN when it is in the low position. BEEP commands are legal at any point. Give a state diagram for this system, in which each edge corresponds to receiving a single command. There should be no edges leading out of the ERROR end state. Explain any notation or decisions that might be unclear.

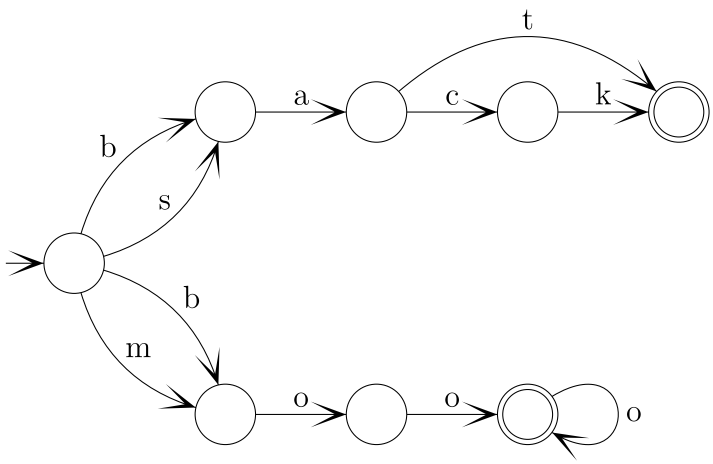
Solution:



(c) Recall that a phone lattice is a type of state diagram, i.e. a directed graph where each node represents a state of a system. A phone lattice has exactly one letter on each edge. Each path from the start node to a final/end node represents a word. Draw a phone lattice representing exactly the following set of words, using a single start node and no more than 9 nodes total.

```
bat, back, sat, sack
boo, booo, boooo, ...    [i.e. b followed by two or more o's]
moo, mooo, moooo, ...    [i.e. m followed by two or more o's]
```

Solution:



11. **Countability**

Let's define sets A and B as follows:

$A = \{0, 2, 4, 6, 8, 10, 12, \ldots\}$, i.e. the even numbers starting with 0.
$B = \{1, 4, 9, 16, 25, 36, 49, \ldots\}$, i.e. perfect squares starting with 1.

Show that $|A| = |B|$ by giving a formula for a specific function $f : A \to B$ that is a bijection. (You do not need to prove that your function f is a bijection.)

Solution: $f(n) = ((n/2) + 1)^2$