CSE 21 Winter 2016
Homework 6
Due: Wednesday, May 11, 2016 at 11:59pm

INSTRUCTIONS

Homework should be done in groups of **one to three** people. You are free to change group members at any time throughout the quarter. Problems should be solved together, not divided up between partners. A **single representative** of your group should submit your work through Gradescope. Submissions must be received by 11:59pm on the due date, and there are no exceptions to this rule.

Homework solutions should be neatly written or typed and turned in through **Gradescope** by 11:59pm on the due date. No late homeworks will be accepted for any reason. You will be able to look at your scanned work before submitting it. Please ensure that your submission is legible (neatly written and not too faint) or your homework may not be graded.

Students should consult their textbook, class notes, lecture slides, instructors, TAs, and tutors when they need help with homework. Students should not look for answers to homework problems in other texts or sources, including the internet. Only post about graded homework questions on Piazza if you suspect a typo in the assignment, or if you don't understand what the question is asking you to do. Other questions are best addressed in office hours.

Your assignments in this class will be evaluated not only on the correctness of your answers, but on your ability to present your ideas clearly and logically. You should always explain how you arrived at your conclusions, using mathematically sound reasoning. Whether you use formal proof techniques or write a more informal argument for why something is true, your answers should always be well-supported. Your goal should be to convince the reader that your results and methods are sound.

For questions that require pseudocode, you can follow the same format as the textbook, or you can write pseudocode in your own style, as long as you specify what your notation means. For example, are you using "=" to mean assignment or to check equality? You are welcome to use any algorithm from class as a subroutine in your pseudocode. For example, if you want to sort list A using InsertionSort, you can call InsertionSort(A) instead of writing out the pseudocode for InsertionSort.

REQUIRED READING Rosen 10.4 through Theorem 1, 11.1, 11.2 through Theorem 2

KEY CONCEPTS Rooted and unrooted trees, binary search trees, basic counting principles (sum and product rules), inclusion-exclusion, quotient (category) rule , coding , coding

---

1. Give an exact answer for each part, and explain the reason for your answer (include justification).

   (a) (2 points) How many ways can you order a burrito if you get to choose from 2 different types of rice, 2 different types of beans, and 5 different types of meat? In addition, you have the option of including corn, sour cream, onions, avocado, and salsa.

   Solution: We use the product rule and multiply the number of options for items in the burrito. For the optional items, there are 2 options each (for example, corn/no corn). This gives $2*2*5*2*2*2*2*2 = 640$.

   (b) (2 points) How many ways can an organization of 10 people elect a president, vice president, and treasurer?

   Solution: There are 10 choices for the president, then 9 choices for the vice president (since we can't pick the person who is president), then 8 choices for the treasurer. This gives $P(10,3) = 10*9*8 = 720$.

   (c) (2 points) How many different strings of length 4 can you make with the letters M,A,M,M,O,T,H?

   Solution: We will use the sum rule to break up the strings of length 4 according to the number of M's they contain.

   If the string contains no M's, it must be made up of the letters A,O,T,H. There are 4! ways to rearrange these four letters, so there are $4! = 24$ strings with no M's.

   If the string contains one M, we must first decide in which of the four positions it should go. There are $\binom{4}{1}$ options for where to place the M. Then the other positions are filled by three of A,O,T,H, and there are $P(4,3) = 4*3*2$ ways to place them. So there are $\binom{4}{1}*4*3*2 = 96$ strings with one M.

   If the string contains two M's, we must first decide in which of the four positions they should go. There are $\binom{4}{2}$ options for where to place the M's. Then the other positions are filled by two of A,O,T,H, and there are $P(4,2) = 4*3$ ways to place them. So there are $\binom{4}{2}*4*3 = 72$ strings with two M's.

   If the string contains three M's, we must first decide in which of the four positions they should go. There are $\binom{4}{3}$ options for where to place the M's. Then the other position is filled by one of A,O,T,H, so there are 4 options. So there are $\binom{4}{3}*4 = 16$ strings with three M's.

   By the sum rule, the total number of strings of length 4 made from the letters M,A,M,M,O,T,H is $24 + 96 + 72 + 16 = 208$.

   (d) (2 points) How many length 8 bitstrings start with 11 or end in 00?

   Solution: We will use inclusion-exclusion since there is some overlap between strings that start with 11 and strings that end in 00. Let $A = \{\text{length 8 bistrings starting with 11}\}$ and $B = \{\text{length 8 bitstrings ending with 00}\}$. Our goal is to compute $|A \cup B|$ using the formula

   $$|A \cup B| = |A| + |B| - |A \cap B|.$$

   For $|A|$, we need to count the number of length 8 bitstrings starting with 11. By the product rule, there is only one option for how to fill in the first two bits (they must be 1s), but 2 options for each of the other 6 bits (they can be 0 or 1). So $|A| = 2^6 = 64$. Similarly, for $|B|$, there is only one option for how to fill in the last two bits (they must be 0s), but two options for each of the other 6 bits so that $|B| = 2^6 = 64$.

   For $|A \cap B|$, we need to count the number of length 8 bitstrings starting with 11 and ending with 00. Now, only the middle four bits have two options, as the first two bits and last two bits are determined. Thus, $|A \cap B| = 2^4 = 16$.

   The inclusion-exclusion formula gives an answer of $64 + 64 - 16 = 112$.

   (e) (2 points) How many length 8 bitstrings do not start with 101?

   Solution: The sum rule says

   $$\begin{aligned}\#\text{ total length 8 bitstrings} &= \#\text{ length 8 bitstrings starting with 101}\\ &\quad + \#\text{ length 8 bitstrings not starting with 101.}\end{aligned}$$

We know the total number of length 8 bitstrings is 2^8 because each bit can be one of 2 options (0 or 1). The number of length 8 bitstrings starting with 101 is 2^5 since the first three bits are determined, but each of the remaining 5 bits can be one of 2 options (0 or 1). Therefore, using the sum rule above, we have

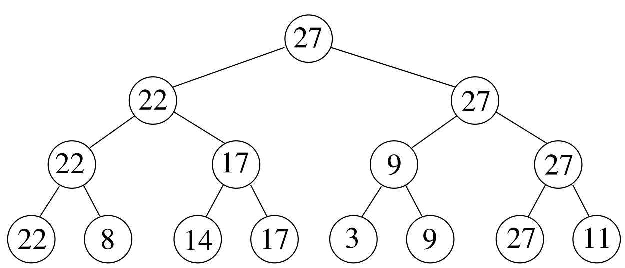
$$\begin{aligned} \#\text{ length 8 bitstrings not starting with 101} &= \#\text{total length 8 bitstrings} \\ &\quad - \#\text{ length 8 bitstrings starting with 101} \\ &= 2^8 - 2^5 \\ &= 224. \end{aligned}$$

(f) (2 points) Spades is a card game played by 4 players with 2 teams of 2. How many ways can the teams be chosen?

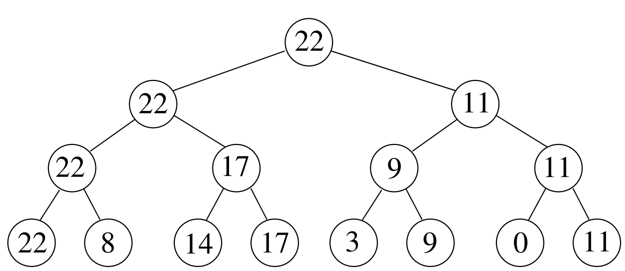
Solution: Say the four players are named A, B, C, and D. To decide on teams, we only have to determine A's partner, then the other two players will be on a team together. There are 3 ways to pick who A will be paired with (B, C, or D) so there are 3 ways to choose teams.

2. A sorting algorithm that uses a binary tree to sort a list of positive integers $a_1, a_2, \ldots, a_n$ from largest to smallest can be described by the following n steps.

Step 1: Construct the binary tree. Output the root value.
The elements $a_1, a_2, \ldots, a_n$ are the leaves of the tree, and we build up the tree one level at a time from there. From left to right, compare the elements in pairs, and put the larger of the two as the parent vertex. Do this at each level until reaching the root, which will be the largest element. Output the value at the root. For example, if the list to be sorted is $22, 8, 14, 17, 3, 9, 27, 11$, the tree would look like this:



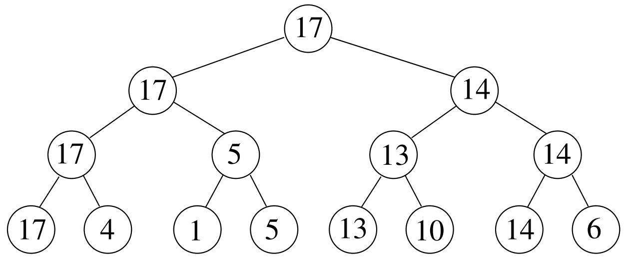
Step 2: Recompute labels. Output the root value.
In the second step, remove the leaf corresponding to that largest element, and replace it with a leaf labeled 0, which is defined to be smaller than all the other list elements. Recompute the labels of all vertices on the path from this 0 to the root. That is, relabel all vertices on the path from the 0 to the root by choosing the larger of the values of their two children. Then the root will become the second-largest element. Output the value at the root. In our example, the tree would now look like this:



Steps 3 through n: Recompute labels. Output the root value.
Repeat the same process as described in step 2. At the end, we will have output the entire list in decreasing order.

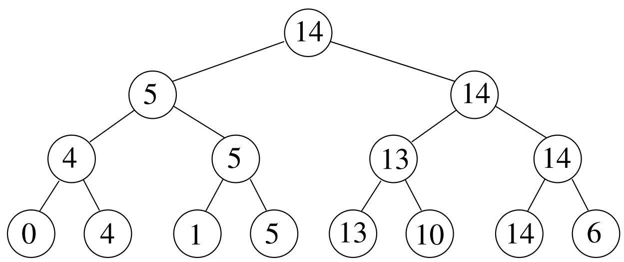
(a) (3 points) Trace through the algorithm as applied to the list $17, 4, 1, 5, 13, 10, 14, 6$. Show the tree at each step.

Solution: Step 1:



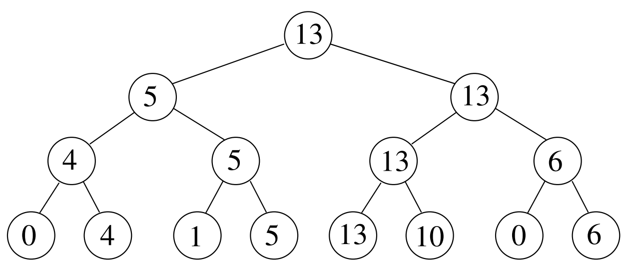
Output 17.

Step 2:



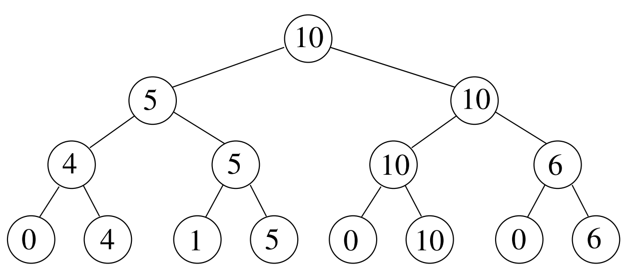
Output 14.

Step 3:



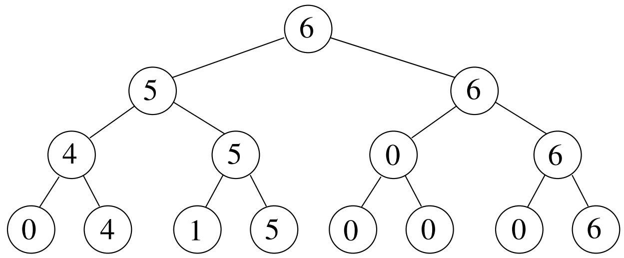
Output 13.

Step 4:



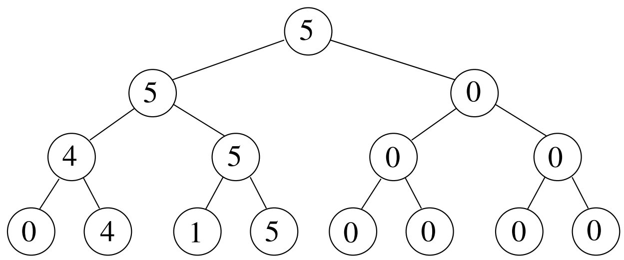
Output 10.

Step 5:



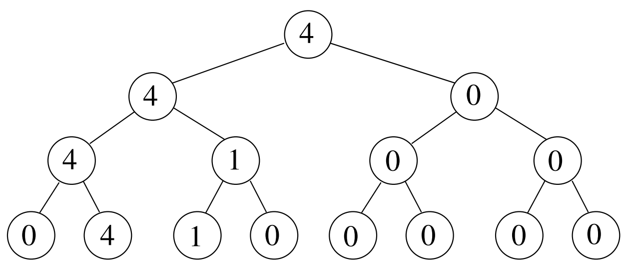
Output 6.

Step 6:



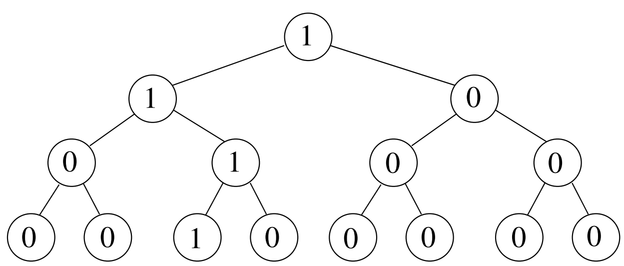
Output 5.

Step 7:



Output 4.

Step 8:



Output 1.

(b) (3 points) If n is a power of two, as in the example, how many comparisons are done at step 1? How many comparisons are done in each of the other steps?

Solution: In the first step, we need to compute the label of all internal vertices of the tree. If there are n leaves, there are $n-1$ internal vertices, so $n-1$ comparisons are done in the first step. At each subsequent step, we only need to recompute the labels of all vertices on a path from the leaves to the root. There is one comparison at each level (excluding the lowest level with leaves), which means we do $\log_2 n$ comparisons at each step besides the first step.

(c) (4 points) If n is a power of two, as in the example, how many comparisons are done throughout the entire algorithm? What is the order of the algorithm in Θ notation?

Solution: We know from part (b) that we do $n-1$ comparisons in the first step, and $\log_2 n$ comparisons at each of the other $n-1$ steps. Therefore, the total number of comparisons throughout the entire algorithm is $(n-1)+(n-1)\log_2 n = n\log_2 n + n - \log_2 n - 1$. In Θ notation, this is $\Theta(n\log_2 n)$ from taking the largest of the individual terms.

3. A *three coloring* of a graph labels each vertex v with one of three *colors*, say R, B or G, so that the two endpoints of any edge have different colors. Consider an undirected graph which is a single path, i.e., where the vertices are $v_1 \ldots v_n$, and there is an edge between each v_i and v_{i+1} for $i = 1..n-1$. How many 3 colorings does this graph have? (2 points correct answer. 1 point short explanation). How many bits are required to describe such a 3-coloring? (2 points correct answer, 1 point short explanation) Give coding and decoding algorithms that given the 3-coloring, outputs a string of the length above that codes it, and given the code, outputs the original 3-coloring. (3 points algorithm description, 1 point short explanation).

4. A normal deck of cards has, for each of 13 possible values (2..10, J,K,Q,A), one card with that face value for each of the four suits (hearts, spades, diamonds, clubs). A can either be the highest or lowest value. Compute the number of five card hands (unordered sets of five distinct cards) with the following properties (2 points each, You can leave your answer in terms of factorials or binomial co-efficients. 1 point correct answer, 1 point explanation):

   (a) The highest value card in the hand is 9 (and aces count higher than 9).
   (b) Two pairs: there are two pairs with the same value, but no three have the same value.
   (c) Flush: All cards have the same suit
   (d) Have at least one card of each suit.
   (e) Have at least one "royal" card (J,K,Q).

5. There are $2n$ people in the tennis club who want to play (individual) tennis. How many different ways are there of pairing them up into n matches so that each member is in exactly one match? You can leave your answer in terms of factorials or binomial co-efficients. (5 points correct answer, 5 points explanation).