

Assignment 3 (V1.01), Due Wednesday, January 27, via Catalyst CollectIt at 11:59 PM

For each of the problems below, show your work, unless otherwise specified.

For the inductive proofs, please format into the following four parts, or points may be deducted: Basis, Induction Hypothesis, Induction Step, Conclusion.

Problem 1 (15 points):

In a tree, a *sibling* is any node that shares its parent with another node. Prove by induction that the number of leaves L in a binary tree T and the number of siblings S in the tree obey the following relationship, provided there is at least one node in the tree:

$$L = (S + 2)/2$$

- (a) **Basis** (3 points). Show that the relationship holds in the case of a one-node tree.
- (b) **Induction Hypothesis** (3 points). We will do induction over the number of nodes in the tree. What is the induction hypothesis?
- (c) **Induction Step** (6 points). Show that if the relationship is true for any tree with n nodes, then it is also true for any tree with $n + 1$ nodes. There are two cases, depending on whether the new node is a sibling or not.
- (d) **Conclusion** (3 points). What is the conclusion?

Problem 2 (15 points):

Let S be a finite set. We define the *power set* of S , denoted $\mathcal{P}(S)$, to be the set of all subsets of S . For example, $\mathcal{P}(\{a, b\}) = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$. Also, the number of elements in S , called the *cardinality* of S , is often denoted $|S|$. For example, here if $S = \{a, b\}$, then $|S| = 2$, and $|\mathcal{P}(S)| = 4$.

- (a) What is $\mathcal{P}(\{1, 2, 3\})$?
- (b) Prove that $|\mathcal{P}(S)| = 2^{|S|}$, for a finite set S , as follows:
 - (i) Let $n = |S|$. How many subsets of size k (where $0 \leq k \leq n$) are there? (Remember that sets are unordered.)
 - (ii) The total number of subsets of S is the sum of the number of subsets of size k for each $0 \leq k \leq n$. Prove that the closed form expression of this sum is 2^n . (*Hint*: use the Binomial Theorem: $(x + y)^n = \sum_{k=0}^n \binom{n}{k} x^k y^{n-k}$, and consider the case when $x = y = 1$.)
- (c) Here, give an alternative combinatorial derivation of (b) along the following lines: since S is finite, number its elements, say $S = \{s_1, \dots, s_n\}$ ($n = |S|$). Let $T \subseteq S$ be any subset and associate to each s_i a binary value: 1 if $s_i \in T$ and 0 otherwise. Using this scheme, how can you represent subsets of S ? From this, how can you deduce (b)?

Problem 3 (8 points):

Use induction to prove that $n! \leq n^n$, for all $n \geq 1$. Remember to show the four parts of an induction proof.

Problem 4 (10 points):

You've asked a colleague to write a fast search method to determine whether a string occurs in a given set of strings. You've told the colleague that the set could be large, and that the method should run efficiently. He's come up with a technique he calls Septenary Search. His method assumes that the input consists of a list L of strings (in sorted order) and a given string s . His method works by calling a HELPER function which is recursive.

The pseudocode for Septenary Search and HELPER is given below.

```
septenarySearch(L, s):
    HELPER(L, 0, L.length-1, s)

HELPER(L, Lo, Hi, s):
    if Hi - Lo + 1 > 7:
        basic_group_size = floor((Hi - Lo + 1) / 7)
        number_of_larger_groups = (Hi - Lo + 1) % 7
        for i in 0 .. 6:
            group_size = basic_group_size
            if i < number_of_larger_groups:
                group_size = group_size + 1
            first_element_of_group = i*group_size
            last_element_of_group = (i+1)*group_size - 1
            if L[first_element_of_group] <= s <= L[last_element_of_group]:
                return HELPER(L, first_element_of_group, last_element_of_group, s)
        else:
            for index in Lo .. Hi:
                if s == L[index]:
                    return true
            return false
```

Prove that the worst case running time of Septenary Search is in $\Theta(\log n)$ by writing a recurrence equation for its running time and solving it. Hint: Note that the number of elements considered in the recursive call to HELPER is at least $1/7$ of those in the range (between Lo and Hi) passed in, and the number is no more than $1/4$ of them.

Problem 5 (10 points):

For each of the three groups of functions below, arrange the functions from slowest growth rate to fastest growth rate, using asymptotic analysis. If any of the functions grow at the same rate, be sure to indicate this. For example, if both $f(n) \in \Theta(n^{13})$ and $g(n) \in \Theta(n^{13})$, then they grow at the same rate.

- a)
- (i) $100n^3$
 - (ii) $n^3 + n$
 - (iii) $n(\log n)^{1000}$

(iv) $\frac{(n+5)!}{(n+4)!}$

b)

(i) $\lceil \log n \rceil$

(ii) $\frac{5}{n}$

(iii) $n \log(n^2)$

c)

(i) $n^{1.5}$

(ii) $2^{n \log n}$

(iii) $2^{(\log n)^9}$

Hint: In some cases taking the log of each of a pair of functions may help to compare them.

Problem 6 (10 points):

Recall that a geometric series can be represented by the general form:

$$\sum_{i=0}^{n-1} ar^i = a + ar + ar^2 + ar^3 + \dots + ar^{n-1}$$

where a is a real constant and r is the common ratio.

The sum of a finite geometric series is given by:

$$\sum_{i=0}^{n-1} ar^i = a \left(\frac{1 - r^n}{1 - r} \right)$$

If the series is infinite and $|r| < 1$, then the geometric series converges to $\frac{a}{1-r}$.

If the series is infinite and $|r| \geq 1$, then the geometric series diverges.

Determine the sum of the following geometric series. If it is infinite and converges, specify the value it converges to. If it converges for only some values of the parameter k , specify the domain for which it converges. (Simplify as much as possible, and round decimals to two places or leave in fractional form.) You may use a graphing calculator or [WolframAlpha](http://www.wolframalpha.com/)¹ for this problem.

a) $\sum_{i=0}^{\infty} 7 \left(-\frac{2}{15}\right)^i$

b) $\sum_{i=0}^{\infty} \left(\frac{10k^2}{2^{k-5}}\right)^i$

c) $\sum_{i=0}^9 \frac{1}{20} \left(\frac{3}{2}\right)^i$

¹<http://www.wolframalpha.com/>

Problem 7 (12 points):

(a) Write $2.5757\overline{57}$ as a fraction in lowest terms. (For example $0.7575\overline{75} = \frac{75}{99}$ and in lowest terms is $\frac{25}{33}$.)

(b) Compute the value of the expression:

$$9.9^2 - \lfloor 9.9 \rfloor \cdot \lceil 9.9 \rceil$$

(c) List all permutations of the sequence [a, b, c].

(d) List all combinations of elements of the set $\{a, b, c\}$ taken 2 at a time. (There should only be 3 of them.)

(e) How many different combinations are there of 3 elements at a time, taken from the set $\{V, W, X, Y, Z\}$?

(f) List all substrings of abcde. You should have 16 of them.

(g) List all subsequences of abcd. You should also have 16 of these.

Problem 8 (10 points):

Let the running time for the following four snippets of code be given by $f_1(n)$, $f_2(n)$, $f_3(n)$, and $f_4(n)$.

For each one, either find a corresponding function $g_1(n)$, $g_2(n)$, $g_3(n)$, or $g_4(n)$, such that $f(n)$ is $\Theta(g(n))$ and g is a “simple” function, or show that no such function exists. “Simple” means it should have no unneeded constants or low-order terms. (E.g., $n^5 \log n$ and x^{17} are simple, but $3n^3$ and $x^2 + x$ are not.)

```
(a) sum = 0
    len = n
    while len > 0:
        for i in range(0, len):
            sum += 1
        len = floor(len / 2)
```

```
(b) sum = 0
    for i in range(0, n):
        for j in range(0, i * n):
            sum += 1
```

```
(c) sum = 0
    for i in range(0, n):
        for j in range(0, i):
            sum += 1
    for k in range(0, 20000):
        sum += 1
```

```
(d) sum = 0
    for i in range(0, n):
        for j in range(0, i**2):
            for k in range(0, 10):
                sum += 1
```

Problem 9 (10 points):

For $h = 0, 1, 2, 3$, and 4 , draw an AVL tree having height h that has as few nodes as possible, and such that whenever two siblings are roots of trees with unequal heights, it is the left subtree that has greater height. (5 points).

Write down the numbers of nodes $F(h)$ in your trees as a function of h for the five trees you have drawn. (3 points).

Give a recurrence equation for $F(h)$ that will let you determine these numbers for trees of greater height without having to first draw the trees and then count their nodes. (2 points)

Problem EC (10 points):

Extra Credit A graph is a set of vertices/nodes V and a set of pairs of vertices E called edges. For our purposes, we assume

- (i) there are no self-loops (edges that go from one node to itself)
- (ii) the edges are undirected (the edges are unordered pairs of vertices)
- (iii) there are no duplicate edges (between any two vertices, there is at most 1 edge connecting them)

Graphs are very useful because they can model many structures such as social networks (people are vertices, and 'friend' relationships are edges) and communication networks (routers and gateways are vertices, and connections are edges). In this problem, you will look at a special type of graph. We call a graph $G = (V, E)$ *fully connected* if for every pair of vertices, there exists an edge between them.

- (a) What does a fully connected graph with 1 vertex look like? 3 vertices? 5 vertices? How many edges are present in each of these graphs? (please draw and write the number next to the corresponding graph)
- (b) Show that in a fully connected graph, we have

$$|E| = \frac{n(n-1)}{2}$$

(*Hint:* Try using induction. You've proven the base case where $n = 1$ in (a). What happens when you add another vertex to an already fully connected graph? How many edges will you then need to add to this new graph to make it fully connected again?)

- (c) Find a purely combinatorial way to prove (b).