Practice Quiz

The Spring '98 6.042 Quiz will be Wednesday, March 18, 7pm–9pm, in rooms **3**-270 and 3-370 (**not** building 4 where the lectures are).

The Quiz is *closed book*; you may however bring a single sheet of paper with personal notes on both sides – ordinary size paper with ordinary size handwriting or type font `:-)`

The Quiz covers the material in Lectures 1–$10\frac{1}{3}$ (through Divide-and-Conquer recurrences), Problem Sets 1–5, and Recitations 1–6.

Problem 1. Propositional Logic

(a) Cast each of the following statements in propositional symbol form.

Definitions:

1. $C =$ set of classes.
2. $M =$ set of majors.
3. for $x \in C$, $H(x)$ means x is hard.
4. for $x \in C, y \in M$, $R(x, y)$ means x is required for y.

Statements:

1. At least one class is hard.

   $$\exists x \in C : H(x)$$

2. Every major requires at least one hard class.

   $$\forall y \in M \exists x \in C : H(x) \wedge R(x, y)$$

3. No class is required for two or more majors.

   $$\neg \exists x \in C, \exists y_1 \in M \exists y_2 \in M : (y_1 \neq y_2) \wedge R(x, y_1) \wedge R(x, y_2)$$

(b) Write the negation of $\forall y \exists x H(x) \wedge R(x, y)$ in terms of $\neg H(x)$ and $\neg R(x, y)$ (and without using any other negation symbols – i.e., $\neg\forall$ is not allowed.)

$$\exists y \forall x \neg H(x) \vee \neg R(x, y)$$

Problem 2. Induction

Use induction to prove that

$$1^3 + 2^3 + \ldots + n^3 = \left(\frac{n(n+1)}{2}\right)^2$$

whenever n is a natural number.

Use induction to prove our proposition, P(n),

$$1^3 + 2^3 + \ldots + n^3 = \left(\frac{n(n+1)}{2}\right)^2$$

whenever n is a natural number.

Base Case: $n = 0$

The left side is an empty sum, so $0 = \left(\frac{0\cdot 1}{2}\right)^2 = 0$.

Inductive Step: Show that $P(n) \to P(n+1)$.

By the inductive hypothesis, we assume that

$$1^3 + 2^3 + \ldots + n^3 = \left(\frac{n(n+1)}{2}\right)^2$$

$$\begin{aligned} 1^3 + 2^3 + \ldots + n^3 + (n+1)^3 & \\ &= \left(\frac{n(n+1)}{2}\right)^2 + (n+1)^3 \\ &= \frac{n^2(n+1)^2 + 4(n+1)(n+1)^2}{4} \\ &= \frac{(n^2+4n+4)(n+1)^2}{4} \\ &= \left(\frac{(n+1)(n+2)}{2}\right)^2 \end{aligned}$$

And our hypothesis is proved.

Problem 3. Induction

Prove the set identity

$((A - B) \cup (B - A))^c = (A^c \cup B) \cap (A \cup B^c)$

Using the definition of set difference, both of De Morgan's Laws, and the commutativity of union (in that order):

$$\begin{aligned}((A-B)\cup(B-A))^c &= ((A\cap B^c)\cup(B\cap A^c))^c\\ &= (A\cap B^c)^c\cap(B\cap A^c)^c\\ &= (A^c\cup B)\cap(B^c\cup A)\\ &= (A^c\cup B)\cap(A\cup B^c)\end{aligned}$$

Alternatively, one can use the definitions of the operations. We need to prove x is in the LHS if and only if it is in the RHS.

$x \in ((A-B)\cup(B-A))^c$ means that $x \notin (A-B)\cup(B-A)$. This means that $x \notin A-B$ and $x \notin B-A$.

So consider cases. If $x \in A$, then certainly $x \in A \cup B^c$. Furthermore, since by assumption $x \notin A-B$, we must have $x \in B$. Thus, $x \in B \cup A^c$. Putting these together shows $x \in (A\cup B^c)\cap(B\cup^c)$. Now consider the other case. If $x \notin A$, meaning $x \in A^c$, then certainly $x \in A^c \cup B$. Also, (since $x \notin B-A$), we must have $x \notin B$. That is, $x \in B^c$. Therefore, $x \in B^c \cup A$. Thus, $x \in (A^c\cup B)\cap(B^c\cup A)$.

This proves one direction; the other is similar.

Problem 4. Induction

Prove that the following two sequences are equal:

$$\begin{aligned}u_0 &= 2\\ u_{n+1} &= u_n^2 - u_n + 1\end{aligned}$$

$$\begin{aligned}v_0 &= 2\\ v_{n+1} &= 1 + v_0 v_1 \cdots v_n\end{aligned}$$

Proof. By strong induction on n.

Base cases: When $n = 0, u_n = v_n$, by definition. When $n = 1$ then we have $u_n = v_n = 3$.

Inductive Step: Assume that the sequences agree up to the kth member, i.e. assume that $u_n = v_n, \forall n, 0 \leq n \leq k$ where $k \geq 1$. Now:

$$\begin{aligned}u_{k+1} &= u_k^2 - u_k + 1\\ &= v_k^2 - v_k + 1 \quad \text{(by the inductive hypothesis)}\\ &= v_k(v_k - 1) + 1\\ &= v_k(v_0 \ldots v_{k-1}) + 1 \quad \text{(by the defininition for } v_k \text{ and the fact that } k \geq 1)\\ &= v_0 \ldots v_k + 1\\ &= v_{k+1}\end{aligned}$$

This implies that the sequences agree up to the $k+1$st element. Thus, by induction, $u_n = v_n, \forall n \geq 0$, i.e. the sequences are equal.
(Note that when we used the formula for v we assumed that k is at least 1. In other words $k+1$ is at least 2. So we can't use this step to go from $n = 0$ to $n = 1$. That's why we had to show that $u_1 = v_1$ as a base case. In fact unless $u_0 = v_0 = 0$ or $u_0 = v_0 = 2$ the sequences aren't the same)

■

Problem 5. Induction
Unbeknownst to many, Leonardo Fibonacci had a brother Eduardo who invented the following sequence of numbers that he called *Eduardo* numbers:

$$e_0 = 1$$
$$e_1 = 3$$
$$\forall i \geq 2 : e_i = e_{i-1} + e_{i-2}$$

Eduardo (who was not so good at math) made 2 conjectures about his numbers:

$$C1 : \forall i \geq 0 : e_i = f_{i+1} + 2f_i$$
$$C2 : \forall i \geq 0 : e_i = f_{i+2} - f_i$$

where f_i is the ith Fibonacci number ($f_0 = 0$, $f_1 = 1$, $f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$).

Decide whether or not Eduardo's conjectures are true. Prove your answers. (Hint: do not repeat Eduardo's mistake by trying to solve the recurrence exactly – there is a much faster way to determine if each conjecture is true.)

The first conjecture can be proven true by induction as follows:

Base cases: $e_o = 1 = f_1 + 2f_0,\ e_1 = 3 = 1 + 2 = f_2 + 2f_1$

Induction step: Suppose $e_i = f_{i+1} + 2f_i$ for $i < k$, where $k \geq 2$. We then have that

$$e_k = e_{k-1} + e_{k-2} = f_k + 2f_{k-1} + f_{k-1} + 2f_{k-2} = f_{k+1} + 2f_k$$

The second conjecture is false. A counterexample is $e_1 = 3 \neq f_3 - f_1 = 1$.

Problem 6. Well Ordering Principle

An infinite sequence of natural numbers $a_1, a_2, a_3, \ldots$ is defined in such a way that $a_1 = 14, a_2 = 35$, and each later a_i, $i \geq 3$, is obtained using one of the following rules:

Rule A Choose any two preceding elements of the sequence and define a_i to be their sum.

Rule B Choose any number $k \geq 1$ of preceding elements of the sequence,and define a_i to be their product divided by 7^{k-1}.

Use the well-ordering principle to prove that every element in the sequence is a multiple of 7. (*Hint: In case you have forgotten, the well-ordering principle says that every nonempty set of naturals has a smallest element.*)

Proof. Let $S = \{$all elements in the sequence that are not multiples of 7 $\}$.
Assume that S is nonempty. Then, by well-ordering, it has a smallest element a_n.
If a_n is obtained by rule A, then $a_n = a_i + a_j$ for some a_i and a_j smaller than a_n. But then a_i and a_j are multiples of 7 therefore a_n has to be a multiple of 7 also.
If a_n is obtained by rule B, then there are k smaller elements in the sequence ,that are multiples of 7, so we can write them in the form $a_k = 7n_k$. a_n is the product of the a_k divided by 7^{k-1}. Then we can write

$$a_n = (\prod_{i=1}^{k}(7n_i))/7^{k-1} = 7(\prod_{i=1}^{k} n_i)$$

Therefore a_n is also a multiple of 7. ■

Problem 7. Graphs

Circle T or F for all questions.

Not all vertices in a 20 vertex tree are leaves. (T)

There is a graph with an odd number of odd degree vertices. (F)

If every vertex in a directed graph has outdegree at least 1, then it has a cycle. (T)

If a graph has more edges than vertices, then it is connected. (F)

If in a simple graph $|E| < |V|/2$, then the graph has a vertex of degree zero. (T)

Problem 8. Graph Coloring

Recall that a coloring of a graph is an assignment of a color to each vertex. A graph is defined to be k-colorable if it has a coloring using at most k colors such that no two adjacent vertices have the same color. (*Note: we consider only graphs without loops in this problem*)

(a) Prove that if a graph has maximum node degree k then it is $(k+1)$-colorable.

Proof by induction on the number of vertices:

Inductive hypothesis:
$P(n) =$"Every graph of n vertices with maximum node degree k is $(k+1)$ colorable."

Base case: $(n = 1)$
The only graph with a single vertex can always be colored with one color, so for any $k \geq 0$, it is $(k+1)$-colorable.

Inductive step:
Assume $P(n)$ in order to prove $P(n+1)$.
Given any graph G_{n+1} with n+1 vertices and maximum degree k, remove one vertex, v, to produce a graph G_n with n vertices. Since we have not added any edges, G_n must have maximum vertex degree at most k. By the inductive hypothesis, we can color G_n with $k+1$ colors. We can then add v back into the graph. Since v has at most k neighbors, we can color it with one of the original $k+1$ colors, giving a $(k+1)$-coloring for G_{n+1}.

(b) A graph is defined to be regular if every vertex in the graph has the same degree. Show that if a connected graph has maximum node degree k and is not regular then it is k-colorable. (*You may use the result from part (a) without proof*).

Proof by strong induction on the number of vertices:

Inductive hypothesis:
$P(n) =$"Every connected graph of n vertices and maximum node degree k that is not regular is k-colorable."

Base case: (n=1)
The graph with one vertex is always regular, so $P(1)$ holds vacuously.

Inductive step:
Assume $P(1) \land P(2) \land \cdots \land P(n)$, in order to prove $P(n+1)$.
Given any graph G_{n+1} with $n+1$ vertices and maximum vertex degree k, $P(n+1)$ holds trivially if the graph is not connected or is regular. Therefore assume G_{n+1} is connected and not regular. Then, since G_{n+1} is not regular, it must have some vertex of degree less than k. Remove that vertex, v, to produce a graph G_n. Then we can consider each connected component C_i of G_n:

Case 1: C_i is regular. Since G_{n+1} was connected, v must have been connected to at least one node in C_i. Therefore, at least one node of C_i must have a lower degree than it did before v was removed. Therefore C_i has at least one node of degree less than k. Since C_i is regular, all nodes must therefore have degree less than k, meaning C_i has a maximum node degree less than k. Therefore C_i is k-colorable by part (a).

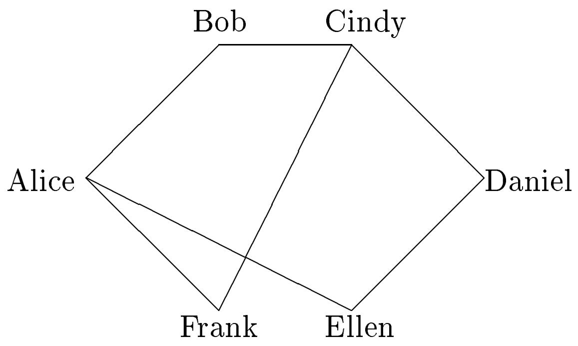
Case 2: C_i is not regular. C_i must have degree at most k and it is connected by definition, so by the inductive hypothesis, it is k-colorable.

Since all components of G_n are k-colorable, G_n is k-colorable. We can therefore assume G_n is assigned some k-coloring. Now we can reattach v. Since v has less than k neighbors, there must be some $j \leq k$ such that v is not adjacent to any node of color j. Thus, we can color v with color j, producing a k-coloring of G_{n+1}. Therefore, if G_{n+1} has maximum vertex degree k and is not regular then it is k-colorable.

Problem 9. Graph Coloring

Alice, Bob, Cindy, Daniel, Ellen, and Frank work at MIT. Alice works Monday, Tuesday, and Saturday. Bob works Tuesday and Wednesday. Cindy works Sunday, Wednesday, and Thursday. Daniel works Thursday and Friday. Ellen works Monday and Friday. Frank works Saturday and Sunday. We want to assign each worker a single parking space at MIT such that at most one person uses each parking space on a given day. Of course, people who work different days can share a parking space.

(a) Draw a conflict graph capturing the constraints on who can share parking spaces.



(b) Give an assignment of people to spaces that uses the fewest possible parking spaces.

There are a variety of solutions, all involving three-colors (or parking spots), one is:
Alice and Cindy
Bob, Frank, and Daniel
Ellen

(c) Prove that you cannot do better.

To do better, we'd have to be able to color the graph with two colors, but this would mean the graph is a bipartite. Since it has an odd cycle (ABCDE), it cannot be a bipartite, therefore 3 colors is the best we can do.

Problem 10. Graph: Tree

For a tree T, let $L(T)$ denote the number of leaves in T and let $H(T)$ be the number of vertices whose degree **exceeds** 2. Prove by induction on the number of vertices in T that $L(T) \geq H(T)$.

Proof by induction the number of vertices, n, in the tree.

We'll start with a base case of 2 vertices since otherwise we need to be careful about degree 0 vertices.

Base case: 2 vertices. Graph is a single edge.
$L(T_1) = 2, H(T_1) = 0$, and $L(T_1) \geq H(T_1)$.

Inductive hypothesis: $P(n)$: $L(T) \geq H(T)$ for any tree of size n or fewer vertices.

Show $P(n) \to P(n+1)$
Given a tree, T_{n+1}, of size $n+1$ vertices. There must exist a leaf (by a theorem shown in class), so remove a leaf. The new tree, T_n, must have $L(T_n) \geq H(T_n)$ by the inductive hypothesis. Now, by removing the vertex from T_{n+1}, the node went from a state of having X edges to Y edges:

- >3 edges to ≥ 3 edges

  $H(T_{n+1}) = H(T_n), L(T_{n+1}) = L(T_n) + 1$, so

  $$H(T_{n+1}) = H(T_n) \leq L(T_n) = L(T_{n+1}) - 1 \leq L(T_{n+1})$$

- 3 edges to 2 edges

  $H(T_{n+1}) = H(T_n) + 1$ and $L(T_{n+1}) = L(T_n) + 1$ so

  $$H(T_{n+1}) = H(T_n) + 1 \leq L(T_n) + 1 = L(T_{n+1})$$

  so $H(T_{n+1}) \leq L(T_{n+1})$.

- 2 edges to 1 edge

  $H(T_{n+1}) = H(T_n)$, and $L(T_{n+1}) = L(T_n) + 1$, so

  $$H(T_{n+1}) = H(T_n) \leq L(T_n) = L(T_{n+1}) - 1 \leq L(T)$$

- 1 edge to no edges
  n must be 2 in this case. $H(T_2) = 0 \leq L(T_2) = 2$, so $0 = H(T_2) \leq L(T_2) = 2$.

Comments:

1. it is easier for this problem to remove a leaf than an internal edge. This is often the case.
2. A lot of people didn't notice that when you remove a leaf, you might create a new leaf.

Problem 11. State Machine

Consider the following algorithm, which takes natural numbers a and b, with $b > 0$, and returns an ordered pair of natural numbers.

Algorithm Div(a, b)

if $(a < b)$
    return $(0, a)$
else
    $(q, r) \leftarrow$Div$(a - b, b)$
    return $(q + 1, r)$

(a) Prove Div terminates in at most $a + 1$ steps.

We model the algorithm as a state machine with state (q, r). The initial state is (a, b). Termination occurs when the termination condition $(a < b)$ is reached.

We define the derived variable $f(q, r) = q$. It is seen that f is non-negative. Furthermore, f is decreasing because then next value of q at each step is $a - b$, where by "step" I mean a recursive call to Div. Therefore, termination occurs in at most a steps ($+1$ for the last iteration).

(b) Prove that Div returns the quotient and remainder of a on division by b.

Recall that the quotient and remainder of a on division by b are defined to be the unique integers q and r such that $a = qb + r$ and $0 \le r < b$. These integers must exist by the Division Algorithm.

We use strong induction on a.

Base case: $a < b$. Then Div immediately returns $(0, a)$. Since $a = 0 \cdot b + a$ and $0 \le a < b$, these are the quotient and remainder of a on division by b.

Inductive step: $a \ge b$. Then $0 \le a - b < a$. In this case, there is a recursive call Div$(a - b, b)$. By the IH, this recursive call returns the quotient and remainder of $a - b$ on division by b. So $a - b = qb + r$ and $0 \le r < b$. Then $a = (q + 1)b + r$ and $0 \le r < b$. So $q + 1$ and r are the quotient and remainder of a on division by b, and these are returned by Div.

Problem 12. State Machine

Consider the following algorithm that takes an array (list) $A[1 \ldots n]$ of numbers such that $A[1] < A[2] < \cdots < A[n]$, and a number q, and is supposed to find q if it is present in the list.

Find(A, n, q)

$i \leftarrow 1$
$j \leftarrow n + 1$
While $(j - i > 1)$
    choose some k with $i < k < j$
    if $A[k] \le q$

$i \leftarrow k$
else
$j \leftarrow k$
Return i

(a) prove this function terminates

Define a state machine whose state is (i, j) and the initial state is $(1, n+1)$. Define the derived variable $f(i, j) = j - i$. Note that at each step we choose a k such that $i < k < j$, and either set $j = k$ or $i = k$. Therefore f is decreasing. Furthermore f is non-negative since $j > i$. We conclude that the algorithm terminates in at most $n + 1 - 1 = n$ steps.

Now suppose that $q = A[r]$ for some r.

(b) State an invariant of the while loop relating i, j, and r. The invariant should be "obviously provable by induction" but you need not prove it. Our invariant is. $i \leq r < j$.

(c) Assuming the invariant, prove that Find returns r.

When the loop terminates, $j - i \leq 1$ (from $j \leq i + 1$). Also, the invariant still holds. So $i \leq r < j \leq i + 1$. Since i and r are integers, $i = r$. Since i is then returned, Find returns r.

Problem 13. State Machine

Luke finished his lightsaber practice, and sits down for a game of 4 dimensional chess. The space of moves is a four dimensional $11 \times 11 \times 11 \times 11$ grid with cells $\{(x, y, z, w) | -5 \leq x, y, z, w \leq 5; x, y, z, w \in \mathbb{Z}\}$. If his king moves 1 unit in each of two different directions at each step, and he starts at $(1, 0, 0, 0)$, can his king reach $(5, 4, -1, 2)$? Express the possible moves as a state machine and find an invariant to prove your answer.

The state is the possible positions (x, y, z, w) on the chess-board $-5 \leq x, y, z, w \leq 5; x, y, z, w \in \mathbb{Z}$. The transitions are the movements of the king, i.e., exactly two coordinates change by ± 1 in a way such that the king remains in the chess board. The initial state is $(1, 0, 0, 0)$.

The invariant is $P(x, y, z, w) : x + y + z + w$ is odd

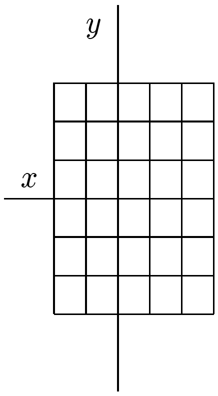
The invariant is certainly true for the initial state, $(1, 0, 0, 0)$.

Assume that it is true at state (x, y, z, w). At the next state transition to state (x', y', z', w') two of the state variables are changed by ± 1, therefore the sum of the invariant is changed by either $-2, 0$, or 2. Therefore $(x' + y' + z' + w')$ remains odd. The invariant is proved.

The state $(5, 4, -1, 2)$ does not satisfy the invariant, therefore it is not a reachable state.

Problem 14. State Machine

David is located at position $(1, 0)$ on the two-dimensional $(\mathbb{Z} \times \mathbb{Z})$ x-y plane. His house is located at the origin. David wants to go home. But it is windy, and each time David takes a one-unit step east or west, he also gets blown a one-unit step north or south (he can pick which).



(a) On the grid above, mark with X's all the points that David can reach in one step. Mark with O's the points that David can reach in two steps. (The axes labeled x and y intersect at David's house.)

(b) Give an informal explanation for why David can never get home.

David starts off with the sum of his coordinates $(1 + 0 = 1)$ odd. Every time he moves, the parity of each coordinate changes, so the sum of his coordinates remains odd. Since the sum of the coordinates of his home $(0 + 0 = 0)$ is even, he can never reach it.

(c) Describe a state machine that represents David's motion, and list its transitions.

We introduce state variables $x, y \in \mathbb{Z}$. We designate exactly one start state: the one for which $x = 1$ and $y = 0$.

The transitions are $(x, y) \to (x \pm 1, y \pm 1)$.

(d) Prove that David can never get home.

We use the following invariant: $x + y$ is odd for all reachable states. This is true for the start state $(0 + 1 = 1)$. For each transition, the sum is modified either by 0, -2, or 2. Thus each transition increases $x + y$ by an even amount preserving the invariant.

Since $x+y$ is even for the state corresponding to home, it is not a reachable state of our state machine. Thus David can never get home.

Problem 15. Summation

Your answer in the following problems should not contain $\sum$ or $\prod$ and should be in closed form.

(a) [**4 points**]
Find tight asymptotic (Θ) bounds for

$$\sum_{k=1}^{n} \frac{1}{\sqrt{k}}$$

We use the integration method for bounding summations.

$$\begin{aligned}
\int_1^n \frac{1}{\sqrt{x}}dx &\le \sum_{k=1}^{n} \frac{1}{\sqrt{k}} &\le 1 + \int_1^n \frac{1}{\sqrt{x}}dx \\
2x^{\frac{1}{2}}|_1^n &\le \sum_{k=1}^{n} \frac{1}{\sqrt{k}} &\le 1 + 2x^{\frac{1}{2}}|_1^n \\
2\sqrt{n} - 2 &\le \sum_{k=1}^{n} \frac{1}{\sqrt{k}} &\le 2\sqrt{n} - 1
\end{aligned}$$

That is

$$\sum_{k=1}^{n} \frac{1}{\sqrt{k}} = \Theta(\sqrt{n}).$$

(b) [**6 points**]
Evaluate the following product

$$\prod_{i=1}^{\infty} e^{(2^{-i})}$$

$$\begin{aligned}
\prod_{i=1}^{\infty} e^{(2^{-i})} &= e^{\sum_{i=1}^{\infty} 2^{-i}} \\
&= e^1 \\
&= e
\end{aligned}$$

Problem 16. Summation, Asymptotic Notation

Short analyses. No need to show your work. **Be careful** to look at the summation indices.

(a) What is the product of the first n odd powers of two: $\Pi_{k=1}^{n} 2^{2k-1}$?

$$\Pi_{k=1}^{n} 2^{2k-1} = 2^{\sum_{k=1}^{n} 2k-1} = 2^{2\sum_{k=1}^{n} k - \sum_{k=1}^{n} 1} = 2^{n(n+1)-n} = 2^{n^2}$$

(b) What is $\sum_{k=0}^{\infty} (1/3)^k$?

$$\sum_{k=0}^{\infty} (1/3)^k = \lim_{k \to \infty} \frac{1 - (\frac{1}{3})^{k+1}}{1 - \frac{1}{3}} = \frac{1}{\frac{2}{3}} = \frac{3}{2}$$

(c) What is $\sum_{k=a}^{b} k$?

$$\sum_{k=a}^{b} k = \sum_{k=1}^{b} k - \sum_{k=1}^{a-1} k = \frac{b(b+1)}{2} - \frac{a(a-1)}{2}$$

(d) Give asymptotic (Θ) bounds for $\sum_{k=n/2}^{n} 1/k$.

$$\sum_{k=n/2}^{n} \frac{1}{k} = \Theta(\int_{n/2}^{n} \frac{1}{k} dk)$$

$$\int_{n/2}^{n} \frac{1}{k} dk = \log(k) \mid_{n/2}^{n} = \log(\frac{n}{n/2}) = \log(2)$$

Since $\log(2)$ is a constant,

$$\sum_{k=n/2}^{n} 1/k = \Theta(1)$$

(e) Give asymptotic (Θ) bounds for $\sum_{k=n/2}^{n} k^5$.

$$\sum_{k=n/2}^{n} k^5 = \Theta(\int_{n/2}^{n} k^5 dk) = \Theta(\frac{k^6}{6} \mid_{n/2}^{n}) = \Theta(\frac{n^6}{6} - \frac{n^6}{2^6 \cdot 6}) = \Theta(n^6)$$

(f) You open a savings account with an initial deposit of s dollars.

At the end of every year, the bank pays into the account interest of p dollars on every dollar in the account at that time. You then immediately (after the interest is paid) deposit d dollars. Give (but do not solve) a recurrence for the amount of money $M(n)$ in your account right after you make your deposit at the end of the nth year.

$M(n) = (1 + p)M(n - 1) + d$ with the initial condition $M(1) = s$ (or $M(0) = s$).

Problem 17. Divide-and-Conquer Recurrence

For the recurrences (a), (b), (c) below, indicate which expression (1)–(6) describes its order of growth. The base condition is $T(1) = 1$.

(a) $T(n) = 8T(n/2) + n^3$, (b) $T(n) = 4T(n/2) + n$, (c) $T(n) = 2T(n/3) + n$.

(1) $\Theta(n)$, (2) $\Theta(n \lg n)$, (3) $\Theta(n^2)$, (4) $\Theta(n^2 \lg n)$, (5) $\Theta(n^3)$, (6) $\Theta(n^3 \lg n)$.

Answers: (a) – 6, (b) – 3, (c) – 1.

This problem was an application of the master theorem. The first two case are straightforward applications of cases 1 and 2 of the master theorem. The third problem requires one to note that $\log_3 2 < 1$. Then, it follows directly from case 3 of the master theorem.

Problem 18. Divide-and-Conquer Recurrence

For each of the following recurrences, find a $\Theta(\cdot)$ bound for $T(n)$.

In all cases, the base condition is $T(1) = 1$.

(a) $T(n) = 4T(n/2) + n^2$.
We use the master theorem. Since $n^2 = \Theta(n^{\log_2 4})$, case 2 applies. The solution is $T(n) = \Theta(n^2 \log n)$.

(b) $T(n) = 4T(n/3) + n$.
We use the master theorem. Since $n = O(n^{\log_3 4 - \epsilon})$ for $\epsilon = \log_3 4 - 1 > 0$, case 1 applies. The solution is $T(n) = \Theta(n^{\log_3 4})$.

(c) $T(n) = 4T(n/4) + n^2$.
We use the master theorem. Since $n^2 = \Omega(n^{\log_4 4 + \epsilon})$ for $\epsilon = 1$, and $4(n/4)^2 \le c \cdot n^2$ for $c = 1/4 < 1$, case 3 applies. The solution is $T(n) = \Theta(n^2)$.

(d) $T(n) = T(n/2) + T(n/4) + n$.
Clearly, $T(n) = \Omega(n)$. We now show that $T(n) = O(n)$ using the substitution method. Substituting $T(n) = \alpha n$ in the recurrence relation we obtain

$$\alpha n = \alpha n/2 + \alpha n/4 + n.$$

Solving the equation shows that $\alpha = 4$. Since $T(1) = 1 \le 4$, we have $T(n) \le 4n$ for all n.

This proves that $T(n) = \Theta(n)$.

(e) $T(n) = T(\sqrt{n}) + 1$.
Iterating the recurrence we obtain

$$\begin{aligned} T(n) &= T(n^{1/2}) + 1 \\ &= T(n^{1/4}) + 1 + 1 \\ &= T(n^{1/8}) + 1 + 1 + 1 \\ &= T(n^{(1/2^k)}) + k \end{aligned}$$

If $n = 2^{(2^k)}$, after k iterations we have

$$T(n) = T(2^{(2^k)(1/2^k)}) + k = T(2) + k.$$

Therefore, $T(n) = \Theta(k) = \Theta(\lg \lg n)$.

Problem 19. Divide-and-Conquer Recurrence

Give tight (Θ) bounds for $T(n)$. Assume that $T(n)$ is constant for $n \leq 2$.

(a) $T(n) = 7T(n/3) + n^2$ Iterating, we see that

$$\begin{aligned}
T(n) &= n^2 + 7\left(\left(\frac{n}{3}\right)^2 + 7\left(\left(\frac{n}{3^2}\right)^2 + 7\left(\left(\frac{n}{3^3}\right)^2 + \cdots\right)\right)\right) \\
&= n^2 + 7\left(\frac{n}{3}\right)^2 + 7^2\left(\frac{n}{3^2}\right)^2 + 7^3\left(\frac{n}{3^3}\right)^2 + \cdots \\
&= n^2\left(1 + \left(\frac{7}{3^2}\right) + \left(\frac{7}{3^2}\right)^2 + \left(\frac{7}{3^2}\right)^3 + \cdots\right) \\
&= \Theta(n^2).
\end{aligned}$$

since we have a decreasing geometric series.

(b) $T(n) = 4T(n/3) + n$

We may again iterate as in part **(a)** to see that

$$\begin{aligned}
T(n) &= n\left(1 + \left(\frac{4}{3}\right) + \left(\frac{4}{3}\right)^2 + \cdots + \left(\frac{4}{3}\right)^{\log_3 n}\right) \\
&= \Theta\left(n\left(\frac{4}{3}\right)^{\log_3 n}\right) \\
&= \Theta\left(n\,(n)^{\log_3 4/3}\right) \\
&= \Theta\left(n^{1+\log_3 4/3}\right) \\
&= \Theta\left(n^{\log_3 4}\right).
\end{aligned}$$