



Massachusetts Institute of Technology
Department of Aeronautics and
Astronautics
Cambridge, MA 02139

Unified Engineering
Spring 2005
Problem Set #9
Solutions

Problem S7 Solution (Signals and Systems)

1. The convolution is given by

$$y(t) = g(t) * u(t) = \int_{-\infty}^{\infty} g(t - \tau)u(\tau) d\tau$$

Note that $u(\tau)$ is nonzero only for $-3 \leq \tau \leq 0$, and $g(t - \tau)$ is nonzero only for $0 \leq t - \tau \leq 3$, that is, for $-3 + t \leq \tau \leq t$. So there are four distinct regimes:

- (a) $t < -3$
- (b) $-3 \leq t \leq 0$
- (c) $0 \leq t \leq 3$
- (d) $t > 3$

For cases (a) and (d), there is no overlap between $g(t - \tau)$ and $u(\tau)$, so $y(t) = 0$. For case (b), the overlap is for $-3 \leq \tau \leq t$. So

$$\begin{aligned} y(t) &= \int_{-\infty}^{\infty} g(t - \tau)u(\tau) d\tau \\ &= \int_{-3}^t \sin(-2\pi(t - \tau)) \sin(2\pi\tau) d\tau \end{aligned}$$

At this point, we have to do a little trig:

$$\begin{aligned} \sin(-2\pi(t - \tau)) \sin(2\pi\tau) &= \sin(2\pi(\tau - t)) \sin(2\pi\tau) \\ &= [\sin(2\pi\tau) \cos(2\pi t) - \cos(2\pi\tau) \sin(2\pi t)] \sin(2\pi\tau) \\ &= \cos(2\pi t) \sin^2(2\pi\tau) - \sin(2\pi t) \cos(2\pi\tau) \sin(2\pi\tau) \\ &= \cos(2\pi t) \frac{1 - \cos(4\pi\tau)}{2} - \sin(2\pi t) \frac{\sin(4\pi\tau)}{2} \end{aligned}$$

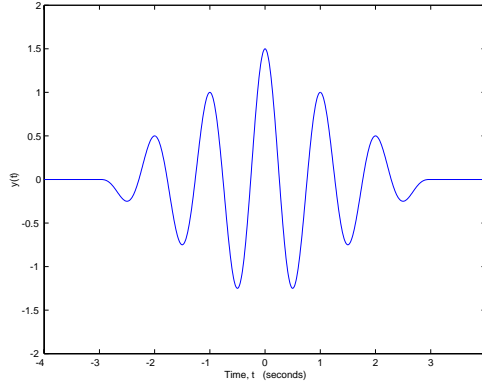
So the integral is given by

$$\begin{aligned} y(t) &= \int_{-3}^t \frac{\cos(2\pi t)}{2} d\tau - \int_{-3}^t \frac{\cos(2\pi t)}{2} \cos(4\pi\tau) d\tau - \int_{-3}^t \frac{\sin(2\pi t)}{2} \sin(4\pi\tau) d\tau \\ &= \frac{\cos(2\pi t)}{2} (t + 3) - \frac{\cos(2\pi t)}{8\pi} \sin(4\pi\tau) \Big|_{\tau=-3}^t + \frac{\sin(2\pi t)}{8\pi} \cos(4\pi\tau) \Big|_{\tau=-3}^t \\ &= \frac{\cos(2\pi t)}{2} (t + 3) - \frac{\cos(2\pi t)}{8\pi} \sin(4\pi t) + \frac{\sin(2\pi t)}{8\pi} [\cos(4\pi t) - 1] \end{aligned}$$

(As often happens with problems involving trig functions, there are other equivalent expressions.) For case (c), the region of integration is $-3 + t \leq \tau \leq 0$. So

$$\begin{aligned} y(t) &= \int_{-3+t}^0 \frac{\cos(2\pi t)}{2} d\tau - \int_{-3+t}^0 \frac{\cos(2\pi t)}{2} \cos(4\pi\tau) d\tau - \int_{-3+t}^0 \frac{\sin(2\pi t)}{2} \sin(4\pi\tau) d\tau \\ &= \frac{\cos(2\pi t)}{2} (3 - t) - \frac{\cos(2\pi t)}{8\pi} \sin(4\pi\tau) \Big|_{\tau=-3+t}^0 + \frac{\sin(2\pi t)}{8\pi} \cos(4\pi\tau) \Big|_{\tau=-3+t}^0 \\ &= \frac{\cos(2\pi t)}{2} (3 - t) + \frac{\cos(2\pi t)}{8\pi} \sin(4\pi t) - \frac{\sin(2\pi t)}{8\pi} [\cos(4\pi t) - 1] \end{aligned}$$

2. $y(t)$ is plotted below.



3. The maximum value of $y(t - T)$ occurs at time T . So I would use this center peak to identify the delay time T .
4. The adjacent peaks are nearly as tall as the center peak, so if noise were added to the signal, the tallest peak might not be the center peak, so we might use the wrong peak to determine the delay time.
5. The chirp signal of Problem S6 produces an ambiguity function with only one prominent peak. Therefore, the addition of noise should not make it difficult to accurately determine the delay time.

Problem S8 Solution (Signals and Systems)

$$1. g(t) = \begin{cases} te^{-at}, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

Therefore,

$$G(s) = \int_0^{\infty} te^{-at}e^{-st} dt$$

Integrate by parts to obtain

$$G(s) = -\frac{t}{s+a}e^{-(a+s)t} \Big|_{t=0}^{\infty} + \frac{1}{s+a} \int_0^{\infty} e^{-at}e^{-st} dt$$

If $\text{Re}[s] > -a$, then the first term evaluates to 0; otherwise, it is undefined. The integral is just the LT of $e^{-at}\sigma(t)$. Therefore,

$$\begin{aligned} G(s) &= \frac{1}{s+a} \int_0^{\infty} e^{-at}e^{-st} dt \\ &= \frac{1}{(s+a)^2}, \quad \text{Re}[s] > -a \end{aligned}$$

$$2. g(t) = \begin{cases} t^2e^{-at}, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

Integrate by parts twice to obtain

$$G(s) = \frac{2}{(s+a)^3}, \quad \text{Re}[s] > -a$$

$$3. g(t) = \begin{cases} t^n e^{-at}, & t \geq 0 \\ 0, & t < 0 \end{cases}, \text{ where } n \text{ is a positive integer.}$$

In general,

$$G(s) = \frac{n!}{(s+a)^{n+1}}, \quad \text{Re}[s] > -a$$

$$4. g(t) = e^{-(t-a)^2/2b^2}, \text{ for all } t.$$

The LT is given by

$$G(s) = \int_{-\infty}^{\infty} e^{-(t-a)^2/2b^2} e^{-st} dt$$

To evaluate the integral, use the fact that we know the integral

$$\int_{-\infty}^{\infty} e^{-u^2/2} du = \sqrt{2\pi}$$

The trick is to put our integral in that form, using a change of variables. The exponent of the LT integral is

$$-(t-a)^2/2b^2 - st = -t^2 - \left(s - \frac{a}{b^2}\right)t + \frac{a^2}{2b^2}$$

We next “complete the square” to obtain

$$-(t - a)^2/2b^2 - st = -(t - a + b^2s)/2b^2 + b^2s^2/2 - as$$

Therefore, the LT integral becomes

$$G(s) = \int_{-\infty}^{\infty} e^{-(t-a+b^2s)^2/2b^2} e^{b^2s^2/2-as} dt$$

Make the change of variables

$$u = \frac{t - a + b^2s}{b}$$
$$du = \frac{dt}{b}$$

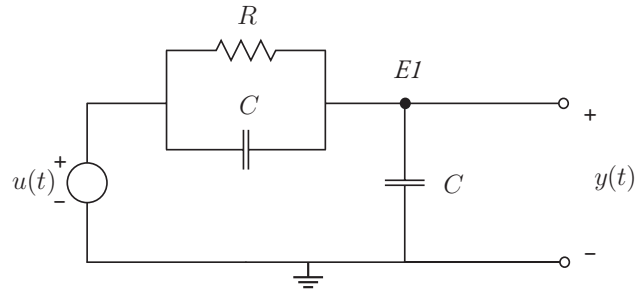
Then the LT integral becomes

$$G(s) = \int_{-\infty}^{\infty} e^{-u^2/2} e^{b^2s^2/2-as} b du$$
$$= \sqrt{2\pi} b e^{b^2s^2/2-as}$$

This LT converges for all s .

Problem S7 Solution (Signals and Systems)

1.



We can use impedance methods to solve for $Y(s)$ in terms of $U(s)$. Label ground and E_1 as shown. Then KCL at E_1 yields

$$Cs(E_1 - 0) + \left(Cs + \frac{1}{R}\right)(E_1 - U) = 0$$

Simplifying, we have

$$\left(2Cs + \frac{1}{R}\right)E_1 = \left(Cs + \frac{1}{R}\right)U$$

Since we are finding the step response,

$$U(s) = \frac{1}{s}, \quad \text{Re}[s] > 0$$

Plugging in numbers, we have

$$(0.2s + 0.5)E_1(s) = (0.1s + 0.5)\frac{1}{s}$$

Solving for E_1 , we have

$$E_1(s) = \frac{0.1s + 0.5}{(0.2s + 0.5)s} = \frac{0.5s + 2.5}{s(s + 2.5)}$$

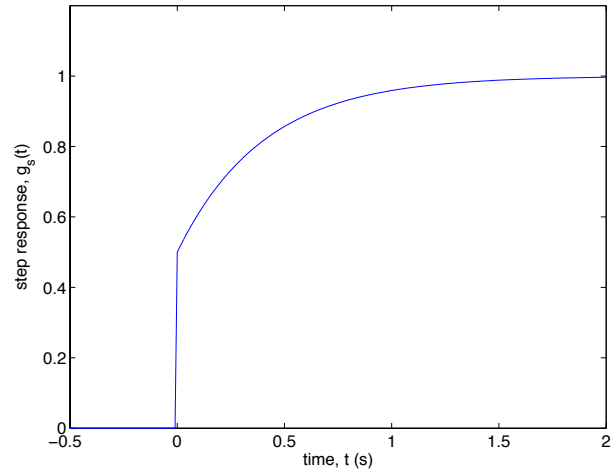
The region of convergence must be $\text{Re}[s] > 0$, since the step response is causal, and the pole at $s = 0$ is the rightmost pole. Using partial fraction expansions,

$$E_1(s) = \frac{1}{s} - \frac{0.5}{s + 2.5}$$

Therefore, $g_s(t) = y(t) = e_1(t)$ is the inverse transform of $E_1(t)$, so

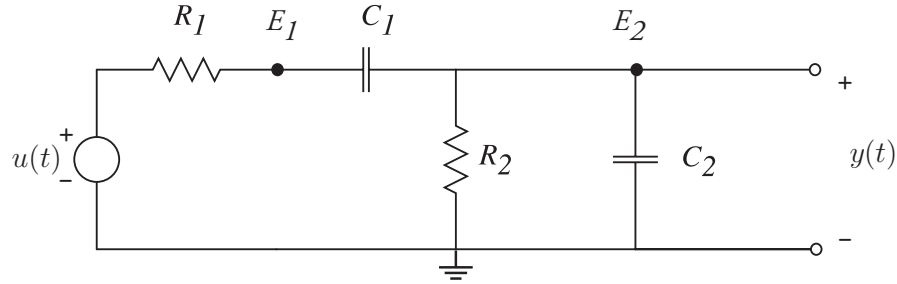
$$y(t) = (1 - 0.5e^{-2.5t})\sigma(t)$$

The step response is plotted below:



Normal differential equation methods are difficult to apply, because we cannot apply the normal initial condition that $e_1(0) = 0$. This is because the chain of capacitors running from the voltage source to ground causes there to be an impulse of current at time $t = 0$, and the voltages across the capacitors change instantaneously at $t = 0$. It is possible to use differential equation methods, we just have to be more careful about the initial conditions. However, Laplace methods are easier.

2.



Again, use impedance methods, using the node labelling above. Then the node equations are

$$\begin{aligned} (C_1s + G_1)E_1 - C_1sE_2 &= G_1U \\ -C_1sE_1 + [(C_1 + C_2)s + G_2]E_2 &= 0 \end{aligned}$$

where $G = 1/R$. We can use Cramer's rule to solve for E_2 :

$$\begin{aligned} E_2(s) &= \frac{\begin{vmatrix} C_1s + G_1 & G_1U(s) \\ -C_1s & 0 \end{vmatrix}}{\begin{vmatrix} C_1s + G_1 & -C_1s \\ -C_1s & (C_1 + C_2)s + G_2 \end{vmatrix}} \\ &= \frac{G_1C_1s}{C_1C_2s^2 + (G_1C_1 + G_1C_2 + G_2C_1)s + G_1G_2} U(s) \end{aligned}$$

Since we are finding the step response,

$$U(s) = \frac{1}{s}, \quad \text{Re}[s] > 0$$

Plugging in numbers, we have

$$Y(s) = E_2(s) = \frac{0.1s}{0.06s^2 + 0.35s + 0.25} \frac{1}{s} = \frac{5/3}{s^2 + 5.833\bar{3}s + 4.166\bar{6}}$$

In order to find $y(t)$, we must expand $Y(s)$ in a partial fraction expansion. To do so, we must factor the denominator, using either numerical techniques or the quadratic formula. The result is

$$s^2 + 5.833\bar{3}s + 4.166\bar{6} = (s + 5)(s + 0.833\bar{3})$$

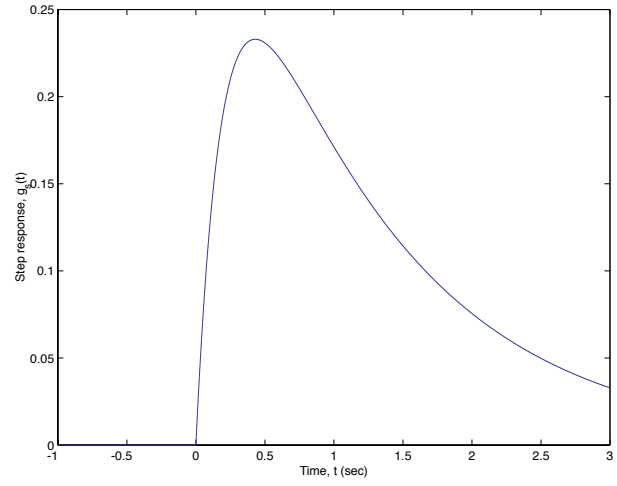
We can use the coverup method to factor $Y(s)$, so that

$$Y(s) = \frac{5/3}{(s + 5)(s + 0.833\bar{3})} = \frac{-0.4}{s + 5} + \frac{0.4}{s + 0.833\bar{3}}$$

The region of convergence must be $\text{Re}[s] > -0.833\bar{3}$, since the step response is causal, and the r.o.c. is to the right of the right-most pole. Therefore, the step response is given by the inverse transform of $Y(s)$, so that

$$g_s(t) = \left(-0.4e^{-5t} + 0.4e^{-0.833\bar{3}t} \right) \sigma(t)$$

The step response is plotted below:



Problem S10 Solution (Signals and Systems)

1. Because the numerator is the same order as the denominator, the partial fraction expansion will have a constant term:

$$\begin{aligned} G(s) &= \frac{3s^2 + 3s - 10}{s^2 - 4} \\ &= \frac{3s^2 + 3s - 10}{(s - 2)(s + 2)} \\ &= a + \frac{b}{s - 2} + \frac{c}{s + 2} \end{aligned}$$

To find a , b , and c , use coverup method:

$$\begin{aligned} a &= G(s)|_{s=\infty} = 3 \\ b &= \left. \frac{3s^2 + 3s - 10}{s + 2} \right|_{s=2} = 2 \\ c &= \left. \frac{3s^2 + 3s - 10}{s - 2} \right|_{s=-2} = 1 \end{aligned}$$

So

$$G(s) = 3 + \frac{2}{s - 2} + \frac{1}{s + 2}, \quad \text{Re}[s] > 2$$

We can take the inverse LT by simple pattern matching. The result is that

$$g(t) = 3\delta(t) + (2e^{2t} + e^{-2t})\sigma(t)$$

- 2.

$$\begin{aligned} G(s) &= \frac{6s^2 + 26s + 26}{(s + 1)(s + 2)(s + 3)} \\ &= \frac{a}{s + 1} + \frac{b}{s + 2} + \frac{c}{s + 3} \end{aligned}$$

Using partial fraction expansions,

$$\begin{aligned} a &= \left. \frac{6s^2 + 26s + 26}{(s + 2)(s + 3)} \right|_{s=-1} = 3 \\ b &= \left. \frac{6s^2 + 26s + 26}{(s + 1)(s + 3)} \right|_{s=-2} = 2 \\ c &= \left. \frac{6s^2 + 26s + 26}{(s + 1)(s + 2)} \right|_{s=-3} = 1 \end{aligned}$$

So

$$G(s) = \frac{3}{s + 1} + \frac{2}{s + 2} + \frac{1}{s + 3}, \quad \text{Re}[s] > -1$$

The inverse LT is given by

$$(3e^{-t} + 2e^{-2t} + e^{-3t})\sigma(t)$$

3. This one is a little tricky — there is a second order pole at $s = -1$. So the partial fraction expansion is

$$G(s) = \frac{4s^2 + 11s + 9}{(s+1)^2(s+2)} = \frac{a}{s+1} + \frac{b}{(s+1)^2} + \frac{c}{s+2}$$

We can find b and c by the coverup method:

$$b = \left. \frac{4s^2 + 11s + 9}{s+2} \right|_{s=-1} = 2$$

$$c = \left. \frac{4s^2 + 11s + 9}{(s+1)^2} \right|_{s=-2} = 3$$

So

$$G(s) = \frac{a}{s+1} + \frac{2}{(s+1)^2} + \frac{3}{s+2}$$

To find a , subtract the second and third terms from above, to obtain

$$\begin{aligned} \frac{a}{s+1} &= G(s) - \frac{2}{(s+1)^2} - \frac{3}{s+2} \\ &= \frac{4s^2 + 11s + 9}{(s+1)^2(s+2)} - \frac{2}{(s+1)^2} - \frac{3}{s+2} \\ &= \frac{4s^2 + 11s + 9 - 2(s+2) - 3(s+1)^2}{(s+1)^2(s+2)} \\ &= \frac{s^2 + 3s + 2}{(s+1)^2(s+2)} \\ &= \frac{1}{(s+1)} \end{aligned}$$

Therefore,

$$G(s) = \frac{1}{s+1} + \frac{2}{(s+1)^2} + \frac{3}{s+2}, \quad \text{Re}[s] > -1$$

The inverse LT is then

$$g(t) = (e^{-t} + 2te^{-t} + 3e^{-2t}) \sigma(t)$$

4. This problem is similar to above. The partial fraction expansion is

$$G(s) = \frac{4s^3 + 11s^2 + 5s + 2}{s^2(s+1)^2} = \frac{a}{s} + \frac{b}{s^2} + \frac{c}{s+1} + \frac{d}{(s+1)^2}$$

We can find b and d by the coverup method

$$b = \left. \frac{4s^3 + 11s^2 + 5s + 2}{(s+1)^2} \right|_{s=0} = 2$$

$$d = \left. \frac{4s^3 + 11s^2 + 5s + 2}{s^2} \right|_{s=-1} = 4$$

So

$$G(s) = \frac{4s^3 + 11s^2 + 5s + 2}{s^2(s+1)^2} = \frac{a}{s} + \frac{2}{s^2} + \frac{c}{s+1} + \frac{4}{(s+1)^2}$$

To find a and c , subtract both terms from both sides, so that

$$\begin{aligned} \frac{a}{s} + \frac{c}{s+1} &= G(s) - \frac{2}{s^2} - \frac{4}{(s+1)^2} \\ &= \frac{4s^3 + 11s^2 + 5s + 2}{s^2(s+1)^2} - \frac{2}{s^2} - \frac{4}{(s+1)^2} \\ &= \frac{4s^3 + 11s^2 + 5s + 2 - 2(s+1)^2 - 4s^2}{s^2(s+1)^2} \\ &= \frac{4s^3 + 5s^2 + s}{s^2(s+1)^2} = \frac{s(4s+1)(s+1)}{s^2(s+1)^2} \\ &= \frac{4s+1}{s(s+1)} = \frac{1}{s} + \frac{3}{s+1} \end{aligned}$$

So

$$G(s) = \frac{1}{s} + \frac{2}{s^2} + \frac{3}{s+1} + \frac{4}{(s+1)^2}$$

and

$$g(t) = (1 + 2t + 3e^{-t} + 4te^{-t}) \sigma(t)$$

5. $G(s)$ can be expanded as

$$\begin{aligned} G(s) &= \frac{s^3 + 3s^2 + 9s + 12}{(s^2 + 4)(s^2 + 9)} \\ &= \frac{s^3 + 3s^2 + 9s + 12}{(s+2j)(s-2j)(s+3j)(s-3j)} \\ &= \frac{a}{s+2j} + \frac{b}{s-2j} + \frac{c}{s+3j} + \frac{d}{s-3j} \end{aligned}$$

The coefficients can be found by the coverup method:

$$\begin{aligned} a &= \left. \frac{s^3 + 3s^2 + 9s + 12}{(s-2j)(s+3j)(s-3j)} \right|_{s=-2j} = 0.5 \\ b &= \left. \frac{s^3 + 3s^2 + 9s + 12}{(s+2j)(s+3j)(s-3j)} \right|_{s=+2j} = 0.5 \\ c &= \left. \frac{s^3 + 3s^2 + 9s + 12}{(s+2j)(s-2j)(s-3j)} \right|_{s=-3j} = 0.5j \\ d &= \left. \frac{s^3 + 3s^2 + 9s + 12}{(s+2j)(s-2j)(s+3j)} \right|_{s=+3j} = -0.5j \end{aligned}$$

Therefore

$$G(s) = \frac{0.5}{s+2j} + \frac{0.5}{s-2j} + \frac{0.5j}{s+3j} + \frac{-0.5j}{s-3j}, \quad \text{Re}[s] > 0$$

and the inverse LT is

$$g(t) = 0.5 (e^{-2jt} + e^{2jt} + je^{-3jt} - je^{3jt}) \sigma(t)$$

This can be expanded using Euler's formula, which states that

$$e^{ajt} = \cos at + j \sin at$$

Applying Euler's formula yields

$$g(t) = (\cos 2t + \sin 2t) \sigma(t)$$

Problem C8. Algorithm Complexity

Compute the complexity of the code shown below using:

- Iteration
- Simplified Master Method (last slide in lecture 9)

```
procedure My_New_Sort(  
    A : in out my_array; left: in integer; right in integer  
    ) is  
    first_split : integer;  
    second_split : integer;  
begin  
    if (left < right) then c1  
        first_split := (left + right) / 3 ; c2  
        second_split := (first_split + right)/2; c2  
        My_New_Sort(A, left, first_split); T(n/3)  
        My_New_Sort(A, first_split+1, second_split); T(n/3)  
        My_New_Sort(A, second_split+1, right); T(n/3)  
        Merge(A, left, first_split, second_split, right); O(n)  
    end if;  
end My_New_Sort;
```

$$\begin{aligned}\text{Therefore } T(n) &= 3T(n/3) + 2c_2 + c_1 + O(n) \\ &= 3T(n/3) + O(n) + C\end{aligned}$$

Solving the recurrence equation using iteration:

$$\begin{aligned}T(n) &= \\ & 3T(n/3) + cn \\ & 3(3T(n/3/3) + cn/3) + cn \\ & 3^2T(n/3^2) + cn3/3 + cn \\ & 3^2T(n/3^2) + cn(3/3 + 1) \\ & 3^2(3T(n/3^2/3) + cn/3^2) + cn(3/3 + 1) \\ & 3^3T(n/3^3) + cn(3^2/3^2) + cn(3/3 + 1) \\ & 3^3T(n/3^3) + cn(3^2/3^2 + 3/3 + 1)\end{aligned}$$

$$\dots$$

$$3^k T(n/3^k) + cn(3^{k-1}/3^{k-1} + 3^{k-2}/3^{k-2} + \dots + 3^2/3^2 + 3/3 + 1)$$

So we have

$$T(n) = 3^k T(n/3^k) + cn(3^{k-1}/3^{k-1} + \dots + 3^2/3^2 + 3/3 + 1)$$

$$\text{For } n = 3^k \quad (k = \log_3 n) T(n) = 3^k T(1) + cn(3^{k-1}/3^{k-1} + \dots + 3^2/3^2 + 3/3 + 1)$$

$$= 3^k c + cn(3^{k-1}/3^{k-1} + \dots + 3^2/3^2 + 3/3 + 1)$$

$$= c3^k + cn(3^{k-1}/3^{k-1} + \dots + 3^2/3^2 + 3/3 + 1)$$

$$= cn3^k/3^k + cn(3^{k-1}/3^{k-1} + \dots + 3^2/3^2 + 3/3 + 1)$$

$$= cn(3^k/3^k + \dots + 3^2/3^2 + 3/3 + 1)$$

So with $k = \log_3 n$

$$T(n) = cn(3^k/3^k + \dots + 3^2/3^2 + 3/3 + 1)$$

$$= cn(k + 1)$$

$$= cn(\log_3 n + 1)$$

$$= \mathbf{O(n \log n)}$$

Solving the recurrence equation using the simplified Master Method:

$$T(n) = 3T(n/3) + 2c_2 + c_1 + O(n)$$

$$= 3T(n/3) + O(n) + C$$

Correlating to the simplified master theorem:

$$cn^k = O(n) \rightarrow k = 1$$

$$aT(n/b) = 3T(n/3)$$

$$\text{Therefore: } T(n) = O(n^k \log_b n) = O(n^1 \log_3 n) = \mathbf{O(n \log_3 n)}$$

Problem C9. Sorting Arrays

Implement the algorithm from part b [merge two consecutive segments of an array] as the merge procedure in the body of the merge sort package:

```
procedure Merge (
    Input_Array : in out My_Array;
    Lb           : in   Integer;
    Mid          : in   Integer;
    Ub          : in   Integer ) is
    -- create a temporary array to store merge sorted values
    Temp        : My_Array := (others => 0);
    L_Index,    :           -- index for incrementing through the lower half of the array
    U_Index,    :           -- index for incrementing through the upper half of the array
    M_Index : Integer;      -- index for incrementing through the temp array
begin

    L_Index := Lb;         -- Set the lower index to the lower bound of the array
    U_Index := Mid+1;     -- Set the upper index to the middle of the array
    M_Index := 1;         -- Set the index for the temporary array to 1

    -- This is what it might look like for an array with 10 elements:
    --
    --           L_Index                U_Index
    -- Input_Array: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
    --
    --           M_Index
    -- Temp:       | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
    --
    -- Think of it as splitting a deck of 10 cards into two decks of 5 cards each.
    -- The lowest valued card between the two decks is removed and put onto the 'Temp' Deck.
    -- This is continued until all 10 cards are on the 'Temp' Deck.

    -- Extrapolating on this analogy:
```



```

-- loop until one of the 5 card decks is gone.
while L_Index <= Mid and U_Index <= Ub loop
  if Input_Array(L_Index) < Input_Array(U_Index) then
    Temp(M_Index) := Input_Array(L_Index);
    L_Index := L_Index + 1;
  else
    Temp(M_Index) := Input_Array(U_Index);
    U_Index := U_Index + 1;
  end if;
  M_Index := M_Index + 1;
end loop;

-- take the cards on the remaining stack and throw them onto the temp stack.
if L_Index > Mid then
  Temp(M_Index..(M_Index+Ub-U_Index)) :=Input_Array(U_Index..Ub);
else
  Temp(M_Index..(M_Index+Mid-L_Index)) :=Input_Array(L_Index..Mid);
end if;

-- Copy the temp array back onto the 'Input_Array'
Input_Array(Lb..Ub) := Temp(1..(Ub-Lb+1));
end Merge;

```

Problem C10. Sorting Linked Lists

Implement insertion sort in the `doubly_linked_list_package` body:

```
procedure Insertionsort (
  L : in out Doublelist ) is
  Temp,
  Pos,
  Current,
  Previous : Doublelistptr;
begin
  -- start at the node after the head. Since you can't insert yourself before yourself.
  Pos := L.Head.Next;

  -- loop through the list, looking at each node.
  while Pos /= null loop
    -- before we start messing with the Pos, get a place holder
    -- by saving the position of the next item to be examined.
    Temp := Pos.Next;

    -- Grab the node out of the list and seal the list.
    if Pos.Next /= null then
      -- If the node is in the middle of the list, just put the two ends together.
      Pos.Next.Prev := Pos.Prev;
      Pos.Prev.Next := Pos.Next;
    else
      -- If the node is at the end of the list, set the end of the list to null;
      Pos.Prev.Next := null;
    end if;
    -- Erase whatever Pos' .Next and .Prev pointers are pointing to.
    Pos.Next := null;
    Pos.Prev := null;
  end loop;
end;
```

```

-- Put the node back into the list.
if Pos.Element <= L.Head.Element then
  -- when the node to be inserted should be the first in the list
  L.Head.Prev := Pos;
  Pos.Next := L.Head;
  L.Head := Pos;
  Pos.Prev := null;
else
  Current := L.Head;
  Previous := null;
  -- when the node to be inserted requires traversal of the list
  while Pos.Element > Current.Element loop
    Previous := Current;
    Current := Current.Next;
    if Current = null then
      exit;
    end if;
  end loop;
  Pos.Next := Current;
  Pos.Prev := Previous;
  Previous.Next := Pos;
  if Current /= null then
    Current.Prev := Pos;
  end if;
end if;

-- restore the next item to be examined
Pos := Temp;
end loop;
end Insertionsort;

```