Mat 2540 HW3

1. Trace Algorithm 1 when it is given $n = 5$ as input. That is, show all steps used by Algorithm 1 to find $5!$, as is done in Example 1 to find $4!$.

ALGORITHM 1 A Recursive Algorithm for Computing $n!$.

procedure *factorial*(n: nonnegative integer)
if $n = 0$ **then return** 1
else return $n \cdot factorial(n - 1)$
{output is $n!$}

Step1: $5! = \text{factorial}(5) = 5 \cdot \text{factorial}(4)$

step2: $\text{factorial}(4) = 4 \cdot \text{factorial}(3)$

step3: $\text{factorial}(3) = 3 \cdot \text{factorial}(2)$

step4: $\text{factorial}(2) = 2 \cdot \text{factorial}(1)$

step5: $\text{factorial}(1) = 1 \cdot \text{factorial}(0)$

and $(0! =)\ \text{factorial}(0) = 1$

$5! = \text{factorial}(5) = 5 \cdot 24 = 120$

$\text{factorial}(4) = 4 \cdot 6 = 24$

$\text{factorial}(3) = 3 \cdot 2 = 6$

$\text{factorial}(2) = 2 \cdot 1 = 2$

$\text{factorial}(1) = 1 \cdot 1 = 1$

3. Trace Algorithm 3 when it finds $\gcd(8, 13)$. That is, show all the steps used by Algorithm 3 to find $\gcd(8, 13)$.

ALGORITHM 3 A Recursive Algorithm for Computing $\gcd(a, b)$.

procedure *gcd*(a, b: nonnegative integers with $a < b$)
if $a = 0$ **then return** b
else return $gcd(b \textbf{ mod } a, a)$
{output is $\gcd(a, b)$}

step1: $\gcd(8, 13) = \gcd(13 \text{ mod } 8, 8) = \gcd(5, 8)$

step2: $\gcd(5, 8) = \gcd(8 \text{ mod } 5, 5) = \gcd(3, 5)$

step3: $\gcd(3, 5) = \gcd(5 \text{ mod } 3, 3) = \gcd(2, 3)$

step4: $\gcd(2, 3) = \gcd(3 \text{ mod } 2, 2) = \gcd(1, 2)$

step5: $\gcd(1, 2) = \gcd(2 \text{ mod } 1, 1) = \gcd(0, 1) = 1$

ALGORITHM 4 Recursive Modular Exponentiation.

procedure *mpower*(b, n, m: integers with $b > 0$ and $m \geq 2, n \geq 0$)
if $n = 0$ **then**
  return 1
else if n is even **then**
  return $mpower(b, n/2, m)^2 \textbf{ mod } m$
else
  return $(mpower(b, \lfloor n/2 \rfloor, m)^2 \textbf{ mod } m \cdot b \textbf{ mod } m) \textbf{ mod } m$
{output is $b^n \textbf{ mod } m$}

5. Trace Algorithm 4 when it is given $m = 5$, $n = 11$, and $b = 3$ as input. That is, show all the steps Algorithm 4 uses to find $3^{11} \textbf{mod } 5$.

step1:
$\text{mpower}(3, 11, 5) \ \dfrac{11 \text{ is odd}}{\text{and } \lfloor \frac{11}{2} \rfloor = 5} \ (\text{mpower}(3, 5, 5))^2 \text{ mod } 5 \cdot (3 \text{ mod } 5)$

$= 3^2 \text{ mod } 5 \cdot 3 \text{ mod } 5$
$= 12 \text{ mod } 5 = \boxed{2}$

step2:
$\text{mpower}(3, 5, 5) \ \dfrac{5 \text{ is odd}}{\lfloor \frac{5}{2} \rfloor = 2} \ (\text{mpower}(3, 2, 5))^2 \text{ mod } 5 \cdot (3 \text{ mod } 5)$

$= (4^2 \text{ mod } 5) \cdot (3 \text{ mod } 5)$
$= 3$

step3:
$\text{mpower}(3, 2, 5) \ \dfrac{2 \text{ is even}}{\lfloor \frac{2}{2} \rfloor = 1} \ (\text{mpower}(3, 1, 5))^2 \text{ mod } 5$

$= 3^2 \text{ mod } 5 = 4$

step4
$\text{mpower}(3, 1, 5) \ \dfrac{1 \text{ is odd}}{\lfloor \frac{1}{2} \rfloor = 0 \quad (n = 0)} \ (\text{mpower}(3, 0, 5))^2 \text{ mod } 5 \cdot (3 \text{ mod } 5) = 1 \cdot 3 \text{ mod } 5 = 3$

$\Rightarrow$ mpower $(3, 11, 5) = 2$

7. Give a recursive algorithm for computing nx whenever n is a positive integer and x is an integer, using just addition.

procedure mult (n: positive integer, x: integer)
If $n=1$ then return x
else return $x + \text{mult}(n-1, x)$

8. Give a recursive algorithm for finding the sum of the first n positive integers.

procedure sum (n: positive integer)
If $n=1$ then return 1
else return $n + \text{sum}(n-1)$

9. Give a recursive algorithm for finding the sum of the first n odd positive integers.

procedure sumofodd (n: positive integer)
If $n=1$ then return 1
else return $(2n-1) + \text{sumofodd}(n-1)$

10. Give a recursive algorithm for finding the maximum of a finite set of integers, making use of the fact that the maximum of n integers is the larger of the last integer in the list and the maximum of the first $n-1$ integers in the list.

procedure max ($a_1, a_2, \cdots, a_n$: integers)
if $n=1$ then return a_1
else return $\max(\max(a_1, a_2, \cdots, a_{n-1}), a_n)$

11. Give a recursive algorithm for finding the minimum of a finite set of integers, making use of the fact that the minimum of n integers is the smaller of the last integer in the list and the minimum of the first $n-1$ integers in the list.

procedure min ($a_1, a_2, \cdots, a_n$. integers)
if $n=1$ then return a_1
else return $\min(\min(a_1, a_2, \cdots, a_{n-1}), a_n)$

29. Devise a recursive algorithm to find the nth term of the sequence defined by $a_0 = 1, a_1 = 2$, and $a_n = a_{n-1} \cdot a_{n-2}$, for $n = 2, 3, 4, \ldots$.

procedure ConsecutiveMulti (n: nonnegative integer)
if n=0 then return 1
else if n=1 then return 2
else return ConsecutiveMulti(n-1) · ConsecutiveMulti (n-2)

30. Devise an iterative algorithm to find the nth term of the sequence defined in Exercise 29.

Procedure iterative ConseMulti (n: nonnegative integer)
if n=0 then return 1
else if n=1 then return 2
else x=1, y=2
for i:=1 to n-1
z:= x·y ; x=y ; y=z
return y.

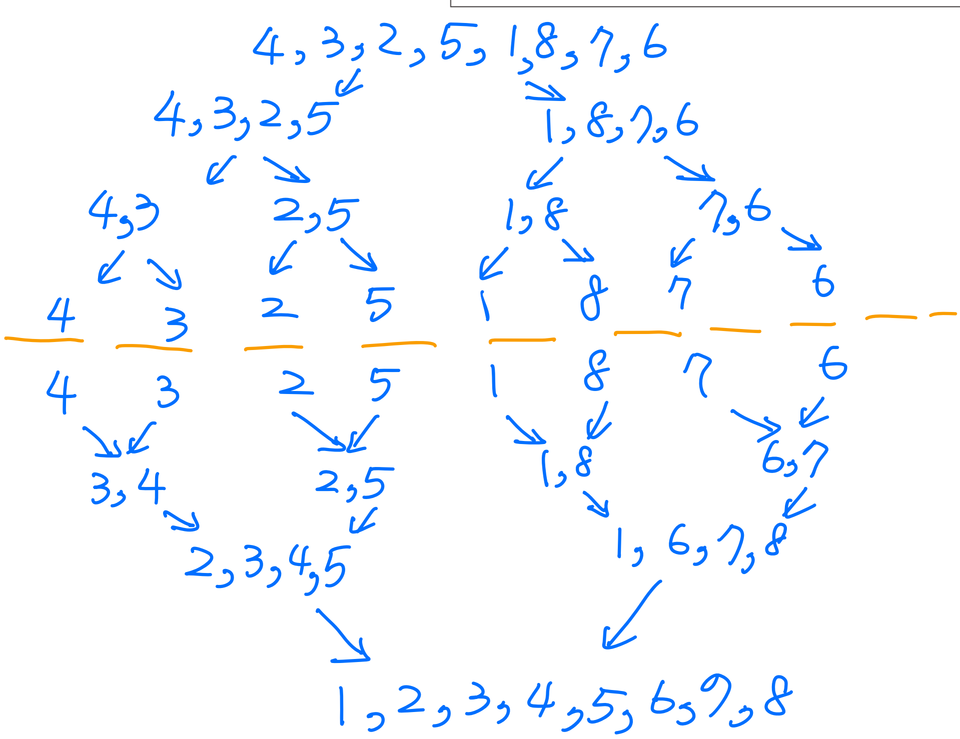
31. Is the recursive or the iterative algorithm for finding the sequence in Exercise 29 more efficient?

iterative one.

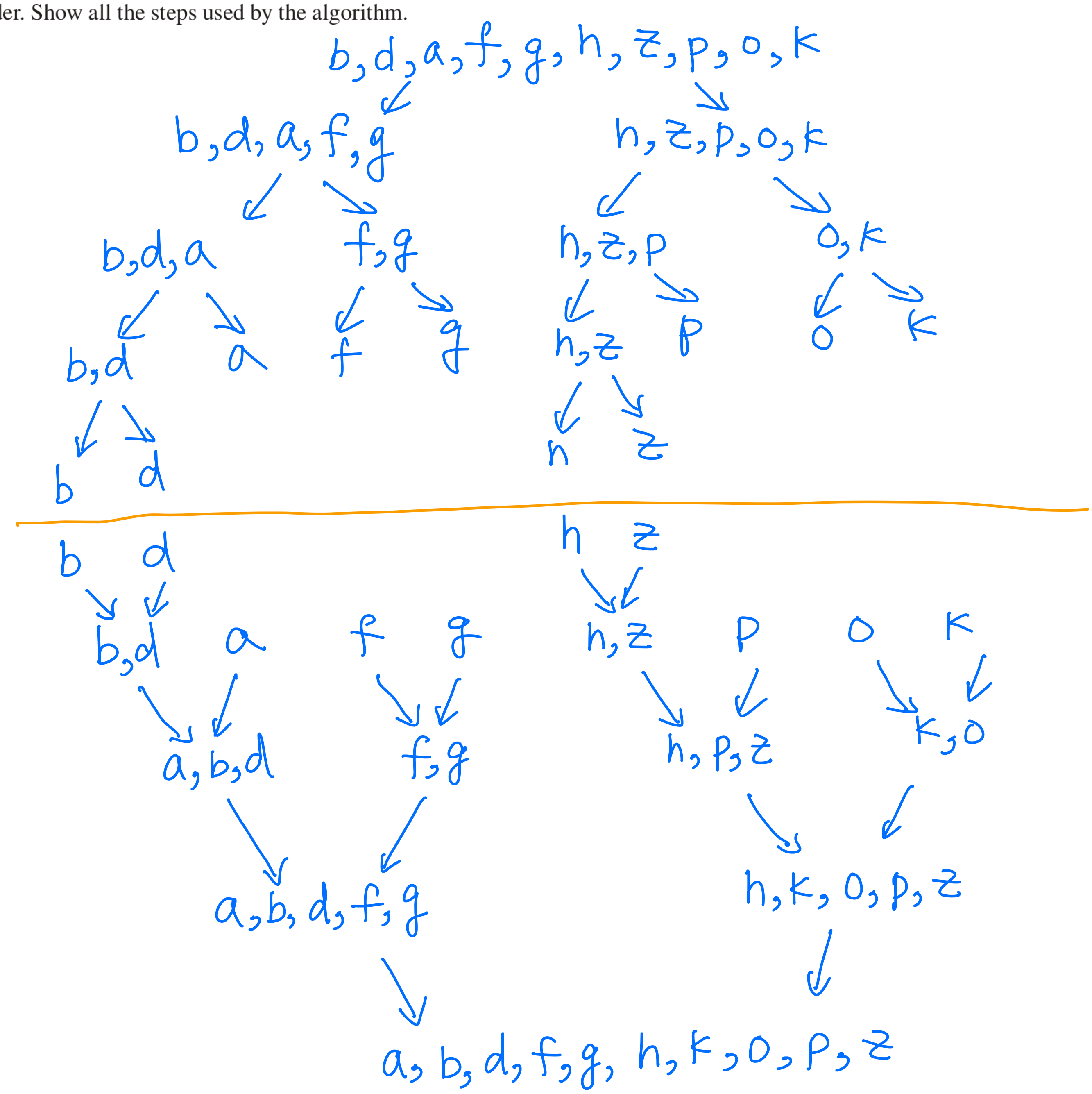
ALGORITHM 9 A Recursive Merge Sort.

procedure *mergesort*($L = a_1, \ldots, a_n$)
if $n > 1$ **then**
$m := \lfloor n/2 \rfloor$
$L_1 := a_1, a_2, \ldots, a_m$
$L_2 := a_{m+1}, a_{m+2}, \ldots, a_n$
$L := merge(mergesort(L_1), mergesort(L_2))$
{L is now sorted into elements in nondecreasing order}

44. Use a merge sort to sort 4, 3, 2, 5, 1, 8, 7, 6 into increasing order. Show all the steps used by the algorithm.



45. Use a merge sort to sort *b, d, a, f, g, h, z, p, o, k* into alphabetic order. Show all the steps used by the algorithm.



46. How many comparisons are required to merge these pairs of lists using Algorithm 10?

a) 1, 3, 5, 7, 9; 2, 4, 6, 8, 10

b) 1, 2, 3, 4, 5; 6, 7, 8, 9, 10

c) 1, 5, 6, 7, 8; 2, 3, 4, 9, 10

ALGORITHM 10 Merging Two Lists.

procedure *merge*(L_1, L_2: sorted lists)
L := empty list
while L_1 and L_2 are both nonempty
  remove smaller of first elements of L_1 and L_2 from its list; put it at the right end of L
  if this removal makes one list empty **then** remove all elements from the other list and append them to L
return L {L is the merged list with elements in increasing order}

a) $L_1 = 1, 3, 5, 7, 9$; $L_2 = 2, 4, 6, 8, 10$; $L = \{\}$

Step1, $1 < 2 \Rightarrow$, $L_1 = 3, 5, 7, 9$, $L_2 = 2, 4, 6, 8, 10$, $L = \{1\}$

Step2, $3 > 2 \Rightarrow L_1 = 3, 5, 7, 9$, $L_2 = 4, 6, 8, 10$, $L = \{1, 2\}$

Step3, $3 < 4 \Rightarrow L_1 = 5, 7, 9$, $L_2 = 4, 6, 8, 10$, $L = \{1, 2, 3\}$

Step4, $5 > 4 \Rightarrow L_1 = 5, 7, 9$, $L_2 = 6, 8, 10$, $L = \{1, 2, 3, 4\}$

Step5, $5 < 6 \Rightarrow L_1 = 7, 9$, $L_2 = 6, 8, 10$, $L = \{1, 2, 3, 4, 5\}$

step6, 7>6 ⇒ $L_1 = 7, 9$, $L_2 = 8, 10$, $L = \{1, 2, 3, 4, 5, 6\}$
step7, 7<8 ⇒ $L_1 = 9$, $L_2 = 8, 10$, $L = \{1, 2, 3, 4, 5, 6, 7\}$
step8, 9>8 ⇒ $L_1 = 9$, $L_2 = 10$, $L = \{1, 2, 3, 4, 5, 6, 7, 8\}$
step9, 9<10 ⇒ $L_1 = \phi$, $L_2 = 10$, $L = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

⇒ since L_1 is empty. We stop here and put L_2 into L. $L = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

⇒ Total 9 comparisons.

(b) $L_1 = 1, 2, 3, 4, 5$, $L_2 = 6, 7, 8, 9, 10$. $L = \phi$

step1: 1<6, $L_1 = 2, 3, 4, 5$, $L_2 = 6, 7, 8, 9, 10$, $L = \{1\}$
step2 2<6, $L_1 = 3, 4, 5$, $L_2 = 6, 7, 8, 9, 10$, $L = \{1, 2\}$
step3 3<6, $L_1 = 4, 5$, $L_2 = 6, 7, 8, 9, 10$, $L = \{1, 2, 3\}$
step4 4<6, $L_1 = 5$, $L_2 = 6, 7, 8, 9, 10$, $L = \{1, 2, 3, 4\}$
step5 5<6, $L_1 = \phi$, $L_2 = 6, 7, 8, 9, 10$, $L = \{1, 2, 3, 4, 5\}$

⇒ since L_1 is empty. we stop here and put L_2 into L
$L = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

⇒ Total 5 comparisons.

(c) $L_1 = 1, 5, 6, 7, 8$, $L_2 = 2, 3, 4, 9, 10$, $L = \phi$

step1. 1<2, $L_1 = 5, 6, 7, 8$, $L_2 = 2, 3, 4, 9, 10$, $L = \{1\}$
step2, 5>2, $L_1 = 5, 6, 7, 8$, $L_2 = 3, 4, 9, 10$, $L = \{1, 2\}$
step3, 5>3, $L_1 = 5, 6, 7, 8$, $L_2 = 4, 9, 10$, $L = \{1, 2, 3\}$
step4, 5>4, $L_1 = 5, 6, 7, 8$, $L_2 = 9, 10$, $L = \{1, 2, 3, 4\}$
step5, 5<9, $L_1 = 6, 7, 8$, $L_2 = 9, 10$, $L = \{1, 2, 3, 4, 5\}$
step6, 6<9, $L_1 = 7, 8$, $L_2 = 9, 10$, $L = \{1, 2, 3, 4, 5, 6\}$
step7, 7<9, $L_1 = 8$, $L_2 = 9, 10$, $L = \{1, 2, 3, 4, 5, 6, 7\}$
step8, 8<9, $L_1 = \phi$, $L_2 = 9, 10$, $L = \{1, 2, 3, 4, 5, 6, 7, 8\}$

⇒ since L_1 is empty, we put L_2 to L and we stop.
$L = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

47. Show that for all positive integers m and n there are sorted lists with m elements and n elements, respectively, such that Algorithm 10 uses $m+n-1$ comparisons to merge them into one sorted list.

Find 2 lists, one has m (L_1) elements and the other (L_2) has n elements, such that once they are merged to a list L by using $m+n-1$ comparisons and $L = \{1, 2, 3, \ldots, m, m+1, m+2, \ldots, m+n\}$.

We can consider $L_1 = \{1, 2, 3, \ldots, m-2, m-1, m+n-1\}$

$L_2 = \{m, m+2, m+3, \ldots, m+n-2, m+n\}$

Then, by Algorithm 10, we have

from step 1 to step $m-1$, we move $1, 2, 3, \ldots, m-1$ from L_1 to L and

$L_1 = \{m+n-1\}$ and $L_2 = \{m, m+2, \ldots, m+n-2, m+n\}$

Here we used $m-1$ comparisons

from step m to step $m+n-2$, we move $m, m+2, \ldots, m+n-2$ from L_2 to L

and $L_1 = \{m+n-1\}$, $L_2 = \{m+n\}$

Here we used $m+n-2-m+1 = n-1$ comparison

at step $m+n-1$, we move $m+n-1$ from L_1 to L and since now L_1 is empty, then we also move $m+n$ from L_2 to L

Here we used 1 comparison

$\Rightarrow$ Total comparison: $(m-1)+(n-1)+1 = m+n-1$

48. What is the least number of comparisons needed to merge any two lists in increasing order into one list in increasing order when the number of elements in the two lists are

a) 1, 4? **b)** 2, 4? **c)** 3, 4? **d)** 4, 4?

We'll come back to this question once we finish the section of tree.