

Name: _____

NetID: _____

Lecture: A B

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

```

01 Jump( $a_1, \dots, a_n$ : an array of  $n$  positive integers)
02 if ( $n = 1$ ) return  $a_1$ 
03 else if ( $n = 2$ ) return  $a_1 + a_2$ 
04 else if ( $n = 3$ ) return  $a_1 + a_2 + a_3$ 
05 else
06      $p = \lfloor n/3 \rfloor$ 
07      $q = \lfloor 2n/3 \rfloor$ 
08     rv = Jump( $a_1, \dots, a_p$ ) + Jump( $a_{p+1}, \dots, a_n$ )
09     rv = rv + Jump( $a_{p+1}, \dots, a_q$ )
10    return rv

```

Dividing an array takes constant time.

1. (5 points) Let $T(n)$ be the running time of Jump. Give a recursive definition of $T(n)$.

Solution:

$$T(1) = a$$

$$T(2) = b$$

$$T(3) = c$$

$$T(n) = 3T(n/3) + d$$

2. (3 points) What is the height of the recursion tree for $T(n)$, assuming n is a power of 3?

Solution: $\log_3(n) - 1$

3. (3 points) What is amount of work (aka sum of the values in the nodes) at level k of this tree?

Solution: $d3^k$

4. (4 points) What is the big-Theta running time of Jump? Briefly justify your answer.

Solution: The number of leaves is $3^{\log_3 n - 1} = \frac{n}{3}$, which is $\Theta(n)$. The total number of nodes is proportional to the number of leaves. Since each node contains a constant amount of work, the running time is proportional to the number of nodes. So the running time is $\Theta(n)$.

Name: _____

NetID: _____

Lecture: A B

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

```

01 Swing(k,n)  \\ inputs are positive integers
02     if (n = 1) return k
03     else if (n = 2) return k2
04     else
05         half = ⌊n/2⌋
06         answer = Swing(k,half)
07         answer = answer*answer
08         if (n is odd)
09             answer = answer*k
10     return answer

```

1. (5 points) Suppose $T(n)$ is the running time of Swing. Give a recursive definition of $T(n)$.

Solution:

$$T(1) = c, T(2) = d$$

$$T(n) = T(n/2) + f$$

2. (4 points) What is the height of the recursion tree for $T(n)$? (Assume that n is a power of 2.)

Solution: $\log_2 n - 1$

3. (3 points) How many leaves are in the recursion tree for $T(n)$?

Solution: One.

4. (3 points) What is the big-Theta running time of Swing?

Solution: $\Theta(\log n)$

Name: _____

NetID: _____

Lecture: A B

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

```

01 Waltz( $a_1, a_2, \dots, a_n$ : list of real numbers)
02 if ( $n = 1$ ) then return 0
03 else if ( $n = 2$ ) then return  $|a_1 - a_2|$ 
04 else
05   L = Waltz( $a_2, a_3, \dots, a_n$ )
06   R = Waltz( $a_1, a_2, \dots, a_{n-1}$ )
07   Q =  $|a_1 - a_n|$ 
08   return max(L,R,Q)

```

Removing the first element of a list takes constant time; removing the last element takes $O(n)$ time.

1. (3 points) Give a succinct English description of what Waltz computes.

Solution: Waltz computes the largest difference between two values in its input list.

2. (4 points) Suppose $T(n)$ is the running time of Waltz. Give a recursive definition of $T(n)$.

Solution: $T(1) = d_1$ $T(2) = d_2$

$$T(n) = 2T(n-1) + cn$$

3. (4 points) What is the height of the recursion tree for $T(n)$?

Solution: We hit the base case when $n - k = 2$, where k is the level. So the tree has height $n - 2$.

4. (4 points) How many leaves are in the recursion tree for $T(n)$?

Solution: 2^{n-2}

Name: _____

NetID: _____ Lecture: A B

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

```

01 Grind( $a_1, \dots, a_n$ )  \\ input is a sorted array of n integers
02     if ( $n = 1$ ) return  $a_1$ 
03     else
04          $m = \lfloor \frac{n}{2} \rfloor$ 
05         if  $a_m > 0$ 
06             return Grind( $a_1, \dots, a_m$ )  \\ constant time to extract part of array
07         else
08             return Grind( $a_{m+1}, \dots, a_n$ )  \\ constant time to extract part of array

```

1. (5 points) Suppose that $T(n)$ is the running time of Grind on an input array of length n and assume that n is a power of 2. Give a recursive definition of $T(n)$.

Solution:

$$T(1) = c$$

$$T(n) = T(n/2) + d$$

2. (4 points) What is the height of the recursion tree for $T(n)$?

Solution: $\log_2 n$

3. (3 points) How many leaves does this tree have?

Solution: One.

4. (3 points) What is the big-Theta running time of Grind?

Solution: $\Theta(\log n)$

Name: _____

NetID: _____

Lecture: A B

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

```

01 Weave( $a_0, \dots, a_{n-1}$ )  \\ input is an array of  $n$  integers ( $n \geq 2$ )
02     if ( $n = 2$  and  $a_0 > a_1$ )
03         swap( $a_0, a_1$ )  \\ interchange the values at positions 0 and 1 in the array
04     else if ( $n > 2$ )
05         p =  $\lfloor \frac{n}{4} \rfloor$ 
06         q =  $\lfloor \frac{n}{2} \rfloor$ 
07         r = p + q
08         Weave( $a_0, \dots, a_q$ )  \\ constant time to make smaller array
09         Weave( $a_{q+1}, \dots, a_{n-1}$ )  \\ constant time to make smaller array
10         Weave( $a_p, \dots, a_r$ )  \\ constant time to make smaller array

```

1. (5 points) Suppose that $T(n)$ is the running time of Weave on an input array of length n . Give a recursive definition of $T(n)$.

Solution:

$$T(2) = d$$

$$T(n) = 3T(n/2) + f$$

2. (4 points) What is the height of the recursion tree for $T(n)$, assuming n is a power of 2?

Solution: $\log_2 n - 1$

3. (3 points) What is the amount of work (aka sum of the values in the nodes) at level k of this tree?

Solution: $f \cdot 3^k$

4. (3 points) How many leaves are in the recursion tree for $T(n)$? (Simplify your answer.)

Solution: $3^{\log_2 n - 1} = 1/3(3^{\log_2 n}) = 1/3(3^{\log_3 n \log_2 3}) = 1/3 \cdot n^{\log_2 3}$

Name: _____

NetID: _____

Lecture: A B

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

```

01 Act( $a_1, \dots, a_n; b_1, \dots, b_n$ ) \\ input is 2 arrays of n integers, n is a power of 2
02     if ( $n = 1$ )
03         return  $a_1 b_1$ 
04     else
05          $p = \frac{n}{2}$ 
06         rv = Act( $a_1, \dots, a_p, b_1, \dots, b_p$ )
07         rv = rv + Act( $a_1, \dots, a_p, b_{p+1}, \dots, b_n$ )
08         rv = rv + Act( $a_{p+1}, \dots, a_n, b_{p+1}, \dots, b_n$ )
09         rv = rv + Act( $a_{p+1}, \dots, a_n, b_1, \dots, b_p$ )
10     return rv

```

1. (5 points) Suppose that $T(n)$ is the running time of Act on an input array of length n . Give a recursive definition of $T(n)$. Assume that dividing an array in half takes constant time.

Solution:

$$T(1) = c$$

$$T(n) = 4T(n/2) + d$$

2. (3 points) What is the height of the recursion tree for $T(n)$, assuming n is a power of 2?

Solution: $\log_2 n$

3. (3 points) What is the amount of work (aka sum of the values in the nodes) at level k of this tree?

Solution: There are 4^k nodes, each containing n . So the total work is $4^k d$

4. (4 points) What is the big-Theta running time of Act. Briefly justify your answer. Recall that $\sum_{k=0}^n a^k = \frac{a^{n+1}-1}{a-1}$.

Solution: The number of leaves is $4^{\log_2 n} = 4^{\log_4 n \log_2 4} = n^{\log_2 4} = n^2$ which is $\Theta(n^2)$. The total number of nodes is proportional to the number of leaves (because $\sum_{k=0}^n 4^k = \frac{4^{n+1}-1}{3}$). Since each node contains a constant amount of work, the running time is proportional to the number of nodes. So the running time is $\Theta(n^2)$.

Name: _____

NetID: _____

Lecture: A B

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

```

01 Dig ( $a_1, \dots, a_n$ : array of integers)
02   if ( $n = 1$ )
03     if ( $a_1 > 8$ ) return true
04     else return false
05   else if (Dig( $a_1, \dots, a_{n-1}$ ) is true and Dig( $a_2, \dots, a_n$ ) is true)
06     return true
07   else return false

```

1. (3 points) If Dig returns true, what must be true of the values in the input array?

Solution: The values in the input array must all be greater than 8.

2. (5 points) Give a recursive definition for $T(n)$, the running time of Dig on an input of length n , assuming it takes constant time to set up the recursive calls in line 05.

Solution:

$$T(1) = c$$

$$T(n) = 2T(n - 1) + d$$

3. (3 points) What is the height of the recursion tree for $T(n)$?

Solution: $n - 1$

4. (4 points) What is the big-theta running time of Dig?

Solution: $\Theta(2^n)$

Name: _____

NetID: _____

Lecture: A B

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

```

01  Swim( $a_1, \dots, a_n$ )  \\ input is a sorted list of n integers
02      if ( $n = 1$ ) return  $a_1$ 
03      else
04           $m = \lfloor \frac{n}{2} \rfloor$ 
05          if  $a_m > 0$ 
06              return Swim( $a_1, \dots, a_m$ )  \\ O(n) time to extract half of list
07          else
08              return Swim( $a_{m+1}, \dots, a_n$ )  \\ O(n) time to extract half of list

```

1. (5 points) Suppose that $T(n)$ is the running time of *Swim* on an input list of length n and assume that n is a power of 2. Give a recursive definition of $T(n)$.

Solution:

$$T(1) = c$$

$$T(n) = T(n/2) + dn$$

2. (4 points) What is the height of the recursion tree for $T(n)$?

Solution: $\log_2 n$

3. (3 points) What value is in each node at level k of this tree?

Solution: $dn/2^k$

4. (3 points) What is the big-Theta running time of *Swim*?

Solution: $\Theta(n)$

[more detail than you need to supply] There is only one node at each level. So the total work is $c + d(n + n/2 + \dots + 2)$. The dominant term of this is proportional to $n \sum_{k=0}^{\log n} 1/2^k = n(2 - 1/2^{\log n}) = n(2 - 1/n) = 2n - 1$.

Name: _____

NetID: _____ Lecture: B

Discussion: Friday 11 12 1 2 3 4

```
1 Jump(A,bottom,top)  \\ A is an array of integers, bottom and top are positive integers
2     if (top = bottom+1) return bottom
3     middle = floor(  $\frac{\text{bottom}+\text{top}}{2}$  )
4     if (A[middle] = 0)
5         return Jump(A, bottom, middle)
6     else
7         return Jump(A, middle, top)
```

1. (3 points) Suppose that A is an array of length n ($n \geq 2$) containing a sequence of positive integers followed by zeros, where $A[1] > 0$ and $A[n] = 0$. What does $\text{Jump}(A,1,n)$ return?

Solution: The location of the last positive value in A.

2. (5 points) Let $T(n)$ be the running time of Jump . Give a recursive definition of $T(n)$.

Solution:

$$T(2) = c$$

$$T(n) = T(n/2) + d$$

3. (3 points) What is amount of work (aka sum of the values in the nodes) at level k in the recursion tree for $T(n)$?

Solution: d

4. (4 points) What is the big-Theta running time of Jump ?

Solution: $\Theta(\log n)$

Name: _____

NetID: _____ Lecture: B

Discussion: Friday 11 12 1 2 3 4

```
01 Skip(k,n)  \\ inputs are natural numbers
02     if (n = 0) return 1
03     else if (n = 1) return k
04     else if (n is odd)
05         temp = Skip(k,floor(n/2))
06         return k*temp*temp
07     else
08         temp = Skip(k,floor(n/2))
09         return temp*temp
```

1. (5 points) Suppose $T(n)$ is the running time of Skip. Give a recursive definition of $T(n)$, assuming that n is a power of 2.

Solution:

$$T(0) = a$$

$$T(1) = c$$

$$T(n) = T(n/2) + d$$

2. (4 points) What is the height of the recursion tree for $T(n)$?

Solution: $\log_2 n$

3. (3 points) How many leaves are in the recursion tree for $T(n)$?

Solution: One.

4. (3 points) What is the big-Theta running time of Skip?

Solution: $\Theta(\log n)$

Name: _____

NetID: _____ Lecture: A B

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

```

01 Hoist( $a_1, \dots, a_n$ : an array of  $n$  positive integers)
02 if ( $n = 1$ ) return 0
03 else if ( $n = 2$ ) return  $a_1 + a_2$ )
04 else
05   p =  $\lfloor n/3 \rfloor$ 
06   q =  $\lfloor 2n/3 \rfloor$ 
07   rv = max(Hoist( $a_1, \dots, a_p$ ), Hoist( $a_{q+1}, \dots, a_n$ ))
08   for i=p to q
09     rv = max(rv,  $a_i + a_{i+1}$ )
10  return rv

```

1. (5 points) Let $T(n)$ be the running time of Hoist. Give a recursive definition of $T(n)$.

Solution:

$$T(1) = b$$

$$T(2) = c$$

$$T(n) = 2T(n/3) + dn$$

2. (3 points) What is the height of the recursion tree for $T(n)$, assuming n is a power of 3?

Solution: $\log_3(n)$

[If n is a power of 3, it will hit the $n = 1$ base case and not the $n = 2$ base case.]

3. (3 points) What is amount of work (aka sum of the values in the nodes) at level k of this tree?

Solution: $\frac{dn}{3^k} 2^k$

4. (4 points) How many leaves does this recursion tree have? Simplify so that your answer is easy to compare to standard running times. Recall that $\log_b x = \log_a x \log_b a$.

Solution: $2^{\log_3 n} = n^{\log_3 2}$

Name: _____

NetID: _____ Lecture: A B

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

```

01 Weave( $a_1, \dots, a_n$ )  \\ input is an array of n integers
02   for  $i = 1$  to  $n - 1$ 
03      $min = i$ 
04     for  $j = i$  to  $n$ 
05       if  $a_j < a_{min}$  then  $min = j$ 
06       swap( $a_i, a_{min}$ )  \\ interchange the values at positions  $i$  and  $min$  in the array

```

1. (3 points) If the input is 10, 5, 2, 3, 8, what are the array values after two iterations of the outer loop?

Solution: After the second iteration, it contains 2, 5, 10, 3, 8.

2. (3 points) Let $T(n)$ be the number of times that line 5 is executed. Express $T(n)$ using summation notation, directly following the structure of the code.

Solution: The i th time through the outer loop, the inner loop runs $n - i + 1$ times. So the total number of times that line 5 executes is:

$$\sum_{i=1}^{n-1} (n - i + 1)$$

3. (3 points) Find an (exact) closed form for $T(n)$. Show your work.

Solution: If we break apart the sum and then substitute in a new index variable $p = n - i$ we get:

$$\sum_{i=1}^{n-1} (n - i + 1) = (n - 1) + \sum_{i=1}^{n-1} (n - i) = (n - 1) + \sum_{p=1}^{n-1} p = (n - 1) + \frac{n(n - 1)}{2}$$

Simplifying, we get

$$(n - 1) + \frac{n(n - 1)}{2} = n - 1 + \frac{1}{2}n^2 - \frac{1}{2}n = \frac{1}{2}n^2 + \frac{1}{2}n - 1$$

4. (3 points) What is the big-theta running time of Weave?

Solution: $\Theta(n^2)$

5. (3 points) Check the (single) box that best characterizes each item.

The running time of Karatsuba's algorithm
is recursively defined by $T(1) = d$ and
 $T(n) =$

$2T(n/2) + cn$

$3T(n/2) + cn$

$4T(n/2) + cn$

$4T(n/2) + c$

Name: _____

NetID: _____ Lecture: A B

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

```

01 Handle( $L_1, L_2$ : sorted lists of integers)
02     if ( $L_1$  is empty)
03         return  $L_2$ 
04     else if ( $L_2$  is empty)
05         return  $L_1$ 
06     else if (head( $L_1$ )  $\leq$  head( $L_2$ ))
07         return cons(head( $L_1$ ), Handle(rest( $L_1$ ),  $L_2$ ))
08     else
09         return cons(head( $L_2$ ), Handle( $L_1$ , rest( $L_2$ )))

```

Assume that head, rest, cons, and testing for the empty list all take constant time.

1. (5 points) Suppose that n is the sum of the lengths of the input lists. Let $T(n)$ be the running time of Handle. Give a recursive definition of $T(n)$.

Solution: $T(1) = T(0) = c$

$$T(n) = T(n - 1) + d$$

2. (3 points) What is the height of the recursion tree for $T(n)$?

Solution: In the worst case, we hit the base case when exactly one of the two lists is empty. That is $n - k = 1$, where k is the level. So the tree has height $n - 1$.

If one list empties while the other still has multiple elements, it's possible for the tree to be shorter. But we're primarily interested in the worst case.

3. (3 points) What is amount of work (aka sum of the values in the nodes) at level k of this tree?

Solution: Notice that the tree doesn't branch, so there is only one node at each level. So the total amount of work at level k is d

4. (4 points) What is the big-theta running time of Handle?

Solution:

$\Theta(n)$ (E.g. unroll the recursive definition.)

Name: _____

NetID: _____ Lecture: A B

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

```

01 Execute( $p_1, \dots, p_n$  : list of  $n$  2D points,  $n \geq 3$ )
02     if ( $n = 3$ )
03         return the largest of  $d(p_1, p_2)$ ,  $d(p_1, p_3)$ , and  $d(p_2, p_3)$ 
04     else
05         x = Execute( $p_2, p_3, p_4, \dots, p_n$ )    \\ removing  $p_1$  from list takes constant time
06         y = Execute( $p_1, p_3, p_4, \dots, p_n$ )    \\ removing  $p_2$  from list takes constant time
07         z = Execute( $p_1, p_2, p_4, \dots, p_n$ )    \\ removing  $p_3$  from list takes constant time
08         return max(x, y, z)

```

The function $d(p, q)$ returns (in constant time) the straight-line distance between two points p and q .

1. (5 points) Suppose $T(n)$ is the running time of Execute on an input array of length n . Give a recursive definition of $T(n)$.

Solution: $T(3) = c$

$$T(n) = 3T(n - 1) + d$$

2. (4 points) What is the height of the recursion tree for $T(n)$?

Solution: We hit the base case when $n - k = 3$, where k is the level. So the tree has height $n - 3$.

3. (3 points) How many leaves are in the recursion tree for $T(n)$?

Solution: 3^{n-3}

4. (3 points) What is the big-Theta running time of Execute?

Solution: $\Theta(3^n)$

Name: _____

NetID: _____ Lecture: A B

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

```

01 Wow(k,n)  \\ inputs are positive integers
02     if (n = 1) return k
03     else
04         half = ⌊n/2⌋
05         answer = Wow(k,half) * Wow(k,half)
06         if (n is odd)
07             answer = answer*k
08     return answer

```

1. (5 points) Suppose $T(n)$ is the running time of Wow. Give a recursive definition of $T(n)$.

Solution:

$$T(1) = c,$$

$$T(n) = 2T(n/2) + d$$

2. (3 points) What is the height of the recursion tree for $T(n)$? (Assume that n is a power of 2.)

Solution: $\log_2 n$

3. (3 points) What is amount of work (aka sum of the values in the nodes) at level k of this tree?

Solution: $d2^k$

4. (4 points) What is the big-Theta running time of Wow?

Solution:

Work at the leaves is $c2^{\log n} = cn$.

Work at internal nodes is $\sum_{k=0}^{\log n-1} d2^k$. This is equal to $d2^{\log n} - 1 = d(n - 1)$.
 $\Theta(n)$

Name: _____

NetID: _____ Lecture: A B

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

```

01 Fabricate( $a_1, \dots, a_n; b_1, \dots, b_n$ )  \\ input is 2 lists of n integers, n is a power of 2
02     if ( $n = 1$ )
03         return  $a_1 b_1$ 
04     else
05          $p = \frac{n}{2}$ 
06         rv = Fabricate( $a_1, \dots, a_p, b_1, \dots, b_p$ )
07         rv = rv + Fabricate( $a_1, \dots, a_p, b_{p+1}, \dots, b_n$ )
08         rv = rv + Fabricate( $a_{p+1}, \dots, a_n, b_{p+1}, \dots, b_n$ )
09         rv = rv + Fabricate( $a_{p+1}, \dots, a_n, b_1, \dots, b_p$ )
10     return rv

```

1. (5 points) Suppose that $T(n)$ is the running time of Fabricate on an input array of length n . Give a recursive definition of $T(n)$. Assume that dividing the list in half takes $O(n)$ time.

Solution:

$$T(1) = c$$

$$T(n) = 4T(n/2) + dn$$

2. (4 points) What is the height of the recursion tree for $T(n)$, assuming n is a power of 2?

Solution: $\log_2 n$

3. (3 points) What is the amount of work (aka sum of the values in the nodes) at level k of this tree?

Solution: There are 4^k nodes, each containing $dn/2^k$. So the total work is $2^k dn$

4. (3 points) How many leaves are in the recursion tree for $T(n)$? (Simplify your answer.)

Solution: $4^{\log_2 n} = 4^{\log_4 n \log_2 4} = n^{\log_2 4} = n^2$

Name: _____

NetID: _____

Lecture: A B

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

```
01 Spin ( $a_1, \dots, a_n$ : array of integers)
02   if ( $n = 1$ )
03     if ( $a_1 > 8$ ) return true
04     else return false
05   else if (Spin( $a_1, \dots, a_{n-1}$ ) is true and Spin( $a_2, \dots, a_n$ ) is true)
06     return true
07   else return false
```

1. (3 points) If Spin returns true, what must be true of the values in the input array?

Solution: The values in the input array must all be greater than 8.

2. (5 points) Give a recursive definition for $T(n)$, the running time of Spin on an input of length n , assuming it takes constant time to set up the recursive calls in line 05.

Solution:

$$T(1) = c$$

$$T(n) = 2T(n - 1) + d$$

3. (3 points) What is the height of the recursion tree for $T(n)$?

Solution: $n - 1$

4. (4 points) What is the big-theta running time of Spin?

Solution: $\Theta(2^n)$

Name: _____

NetID: _____ Lecture: A B

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

```

01 Weave( $a_1, \dots, a_n$ : an array of  $n$  positive integers)
02 if ( $n = 1$ ) return 0
03 else if ( $n = 2$ ) return  $\max(a_1, a_2)$ 
04 else
05    $p = \lfloor n/3 \rfloor$ 
06    $q = \lfloor 2n/3 \rfloor$ 
07    $rv = \max(\text{Weave}(a_1, \dots, a_p), \text{Weave}(a_{q+1}, \dots, a_n))$ 
08   return rv

```

Dividing an array takes constant time.

1. (5 points) Let $T(n)$ be the running time of Weave. Give a recursive definition of $T(n)$.

Solution:

$$T(1) = b$$

$$T(2) = c$$

$$T(n) = 2T(n/3) + d$$

2. (3 points) What is the height of the recursion tree for $T(n)$, assuming n is a power of 3?

Solution: $\log_3(n)$

[If n is a power of 3, it will hit the $n = 1$ base case and not the $n = 2$ base case.]

3. (3 points) What is amount of work (aka sum of the values in the nodes) at level k of this tree?

Solution: There are 2^k nodes at this level, each containing d work. So the total work is $d2^k$.

4. (4 points) How many leaves does this recursion tree have? Simplify so that your answer is easy to compare to standard running times. Recall that $\log_b x = \log_a x \log_b a$.

Solution: $2^{\log_3 n} = n^{\log_3 2}$

Name: _____

NetID: _____ Lecture: A B

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

```

01 Knit( $p_1, \dots, p_n$  : list of  $n$  2D points,  $n \geq 3$ )
02     if ( $n = 3$ )
03         return the largest of  $d(p_1, p_2)$ ,  $d(p_1, p_3)$ , and  $d(p_2, p_3)$ 
04     else
05         x = Knit( $p_2, p_3, p_4, \dots, p_n$ )
06         y = Knit( $p_1, p_3, p_4, \dots, p_n$ ) \ p2 has been removed
07         z = Knit( $p_1, p_2, \dots, p_{n-1}$ )
08         return max(x, y, z)

```

The function $d(p, q)$ returns (in constant time) the straight-line distance between two points p and q . Removing the first/second element of a list takes constant time; removing the last element takes $O(n)$ time.

- (5 points) Suppose $T(n)$ is the running time of Knit on an input array of length n . Give a recursive definition of $T(n)$.

Solution: $T(3) = c$

$$T(n) = 3T(n - 1) + dn$$

- (4 points) What is the amount of work (aka sum of the values in the nodes) at non-leaf level k of this tree?

Solution: At level k , there are 3^k nodes and each node contains $d(n - k)$. So the total work is $3^k(dn - dk)$.

- (3 points) How many leaves are in the recursion tree for $T(n)$?

Solution: 3^{n-3}

- (3 points) Is the running time of Knit $O(2^n)$?

Solution: No, the running time can't be $O(2^n)$. The work in the leaves is $\Theta(3^n)$ and 3^n grows faster than 2^n .

Name: _____

NetID: _____ Lecture: A B

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

```

00 Churn( $a_1, \dots, a_n$ ) : list of  $n$  positive integers,  $n \geq 2$ )
01     if ( $n = 2$ ) return  $|a_1 - a_2|$ 
02     else
03         bestval = 0
04         for  $k = 1$  to  $n$ 
05             newval = Churn( $a_1, a_2, \dots, a_{k-1}, a_{k+1}, \dots, a_n$ )  \\ constant time to remove  $a_k$ 
06             if (newval > bestval) bestval = newval
07         return bestval

```

1. (3 points) Describe (in English) what Churn computes.

Solution: Churn computes the largest difference between two values in the list. Or, equivalently, the largest value minus the smallest value.

2. (5 points) Suppose that $T(n)$ is the running time of Churn on an input list of length n . Give a recursive definition of $T(n)$.

Solution:

$$T(2) = c$$

$$T(n) = nT(n-1) + dn$$

3. (3 points) What is the height of the recursion tree for $T(n)$?

Solution: $n - 2$

4. (4 points) How many leaf nodes are there in the recursion tree for $T(n)$?

Solution: $\frac{n!}{2}$

Name: _____

NetID: _____ Lecture: A B

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

```

01 Grind( $a_1, a_2, \dots, a_n$ : list of real numbers)
02 if ( $n = 1$ ) then return 0
03 else if ( $n = 2$ ) then return  $|a_1 - a_2|$ 
04 else
05   L = Grind( $a_2, a_3, \dots, a_n$ )
06   R = Grind( $a_1, a_2, \dots, a_{n-1}$ )
07   Q =  $|a_1 - a_n|$ 
08   return max(L,R,Q)

```

Removing the first element of a list takes constant time; removing the last element takes $O(n)$ time.

1. (3 points) Give a succinct English description of what Grind computes.

Solution: Grind computes the largest difference between two values in its input list.

2. (4 points) Suppose $T(n)$ is the running time of Grind. Give a recursive definition of $T(n)$.

Solution: $T(1) = d_1$ $T(2) = d_2$

$$T(n) = 2T(n-1) + cn$$

3. (4 points) What is the height of the recursion tree for $T(n)$?

Solution: We hit the base case when $n - k = 2$, where k is the level. So the tree has height $n - 2$.

4. (4 points) How many leaves are in the recursion tree for $T(n)$?

Solution: 2^{n-2}

Name: _____

NetID: _____ Lecture: A B

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

```
01 Sew(k,n)  \\ inputs are natural numbers
02     if (n = 0) return 1
03     else if (n = 1) return k
04     else if (n is odd)
05         temp = Sew(k,floor(n/2))
06         return k*temp*temp
07     else
08         temp = Sew(k,floor(n/2))
09         return temp*temp
```

1. (5 points) Suppose $T(n)$ is the running time of Sew. Give a recursive definition of $T(n)$, assuming that n is a power of 2.

Solution: $T(0) = a$

$T(1) = c$

$T(n) = T(n/2) + d$

2. (4 points) What is the height of the recursion tree for $T(n)$?

Solution: $\log_2 n$

3. (3 points) How many leaves are in the recursion tree for $T(n)$?

Solution: One.

4. (3 points) What is the big-Theta running time of Sew?

Solution: $\Theta(\log n)$

Name: _____

NetID: _____

Lecture: A B

Discussion: Thursday Friday 10 11 12 1 2 3 4 5 6

```

01 Munch( $a_1, \dots, a_n$ : an array of  $n$  positive integers)
02   if ( $n = 1$ ) return  $a_1$ 
03   else if ( $n = 2$ ) return  $a_1 + a_2$ 
04   else if ( $n = 3$ ) return  $a_1 + a_2 + a_3$ 
05   else
06      $p = \lfloor n/3 \rfloor$ 
07      $q = \lfloor 2n/3 \rfloor$ 
08     rv = Munch( $a_1, \dots, a_p$ ) + Munch( $a_{p+1}, \dots, a_q$ )
09     rv = rv + Munch( $a_{q+1}, \dots, a_n$ )
10   return rv

```

Dividing an array takes constant time.

1. (5 points) Let $T(n)$ be the running time of Munch. Give a recursive definition of $T(n)$.

Solution:

$$T(1) = a$$

$$T(2) = b$$

$$T(3) = c$$

$$T(n) = 3T(n/3) + d$$

2. (3 points) What is the height of the recursion tree for $T(n)$, assuming n is a power of 3?

Solution: $\log_3(n) - 1$

3. (3 points) What is amount of work (aka sum of the values in the nodes) at level k of this tree?

Solution: $d3^k$

4. (4 points) What is the big-Theta running time of Munch? Briefly justify your answer.

Solution: The number of leaves is $3^{\log_3 n - 1} = \frac{n}{3}$, which is $\Theta(n)$. The total number of nodes is proportional to the number of leaves. Since each node contains a constant amount of work, the running time is proportional to the number of nodes. So the running time is $\Theta(n)$.

Name: _____

NetID: _____

Lecture: A B

Discussion: Thursday Friday 10 11 12 1 2 3 4 5 6

```

01 Crunch( $a_0, \dots, a_{n-1}$ )  \\ input is an array of  $n$  integers ( $n \geq 2$ )
02     if ( $n = 2$  and  $a_0 > a_1$ )
03         swap( $a_0, a_1$ )  \\ interchange the values at positions 0 and 1 in the array
04     else if ( $n > 2$ )
05          $p = \lfloor \frac{n}{4} \rfloor$ 
06          $q = \lfloor \frac{n}{2} \rfloor$ 
07          $r = p + q$ 
08         Crunch( $a_0, \dots, a_q$ )  \\ constant time to make smaller array
09         Crunch( $a_{q+1}, \dots, a_{n-1}$ )  \\ constant time to make smaller array
10         Crunch( $a_p, \dots, a_r$ )  \\ constant time to make smaller array

```

1. (5 points) Suppose that $T(n)$ is the running time of Crunch on an input array of length n . Give a recursive definition of $T(n)$.

Solution:

$$T(1) = c, T(2) = d$$

$$T(n) = 3T(n/2) + f$$

2. (4 points) What is the height of the recursion tree for $T(n)$, assuming n is a power of 2?

Solution: $\log_2 n - 1$

3. (3 points) What is the amount of work (aka sum of the values in the nodes) at level k of this tree?

Solution: $f \cdot 3^k$

4. (3 points) How many leaves are in the recursion tree for $T(n)$? (Simplify your answer.)

Solution: $3^{\log_2 n - 1} = 1/3(3^{\log_2 n}) = 1/3(3^{\log_3 n \log_2 3}) = 1/3 \cdot n^{\log_2 3}$

Name: _____

NetID: _____

Lecture: A B

Discussion: Thursday Friday 10 11 12 1 2 3 4 5 6

```

01 Crumple( $a_1, \dots, a_n$ : a list of  $n$  positive integers)
02 if ( $n = 1$ ) return  $a_1$ 
03 else if ( $n = 2$ ) return  $a_1 + a_2$ 
04 else if ( $n = 3$ ) return  $a_1 + a_2 + a_3$ 
05 else
06    $p = \lfloor n/3 \rfloor$ 
07    $q = \lfloor 2n/3 \rfloor$ 
08   rv = Crumple( $a_1, \dots, a_p$ ) + Crumple( $a_{q+1}, \dots, a_n$ )
09   rv = rv + Crumple( $a_{p+1}, \dots, a_q$ )
10  return rv

```

Dividing a list takes $O(n)$ time.

1. (5 points) Let $T(n)$ be the running time of Crumple. Give a recursive definition of $T(n)$.

Solution:

$$T(2) = c$$

$$T(n) = 3T(n/3) + dn + f$$

2. (3 points) What is the height of the recursion tree for $T(n)$, assuming n is a power of 3?

Solution: $\log_3(n) - 1$

3. (3 points) What is amount of work (aka sum of the values in the nodes) at level k of this tree?

Solution: There are 3^k nodes, each containing $\frac{dn}{3^k} + f$. So the total work is $dn + f3^k$.

4. (4 points) What is the big-Theta running time of Crumple?

Solution: If we look at the dn part of the node contents, there are $\log_3(n) - 1$ levels, each having n total work. So $\Theta(n \log n)$ total work. (The base of the log changes this only by a constant, therefore does not matter to a big-Theta answer.)

To analyze the $f3^k$ part of the node contents, we need to sum across all the levels. Quick (but ok) version: This sum is proportional to the number of nodes in the lowest layer, which is $f3^{\log_3 n - 2}$, which is proportional to $3^{\log_3 n} = n$ which is $\Theta(n)$.

Since $\Theta(n \log n)$ is the faster growing of these two answers, the running time of Crumple is $\Theta(n \log n)$.

Name: _____

NetID: _____

Lecture: A B

Discussion: Thursday Friday 10 11 12 1 2 3 4 5 6

```

01 Slide( $a_1, \dots, a_n$ )  \\ input is a linked list of n integers
02     if ( $n = 1$ ) return  $a_1$ 
03     else
04          $m = \lfloor \frac{n}{2} \rfloor$ 
05          $p = \text{Slide}(a_1, \dots, a_m)$   \\ O(n) time to split list
06          $q = \text{Slide}(a_{m+1}, \dots, a_n)$   \\ O(n) time to split list
06     return max(p,q)

```

1. (5 points) Suppose that $T(n)$ is the running time of Slide on an input array of length n and assume that n is a power of 2. Give a recursive definition of $T(n)$.

Solution:

$$T(1) = c$$

$$T(n) = 2T(n/2) + dn + f$$

2. (4 points) What is the height of the recursion tree for $T(n)$?

Solution: $\log_2 n$

3. (3 points) What is the amount of work (aka sum of the values in the nodes) at non-leaf level k of this tree?

Solution: There are 2^k nodes, each containing $f + dn/2^k$. So the total is $2^k f + dn$

4. (3 points) What is the big-Theta running time of Slide?

Solution: $\Theta(n \log n)$

Name: _____

NetID: _____ Lecture: A B

Discussion: Thursday Friday 10 11 12 1 2 3 4 5 6

```

01 Swing( $a_1, \dots, a_n; b_1, \dots, b_n$ )  \\ input is 2 arrays of n integers, n is a power of 2
02     if ( $n = 1$ )
03         return  $a_1 b_1$ 
04     else
05          $p = \frac{n}{2}$ 
06         rv = Swing( $a_1, \dots, a_p, b_1, \dots, b_p$ )
07         rv = rv + Swing( $a_1, \dots, a_p, b_{p+1}, \dots, b_n$ )
08         rv = rv + Swing( $a_{p+1}, \dots, a_n, b_{p+1}, \dots, b_n$ )
09         rv = rv + Swing( $a_{p+1}, \dots, a_n, b_1, \dots, b_p$ )
10     return rv

```

1. (5 points) Suppose that $T(n)$ is the running time of Swing on an input array of length n . Give a recursive definition of $T(n)$. Assume that dividing an array in half takes constant time.

Solution:

$$T(1) = c$$

$$T(n) = 4T(n/2) + d$$

2. (3 points) What is the height of the recursion tree for $T(n)$, assuming n is a power of 2?

Solution: $\log_2 n$

3. (3 points) What is the amount of work (aka sum of the values in the nodes) at level k of this tree?

Solution: There are 4^k nodes, each containing n . So the total work is $4^k d$

4. (4 points) What is the big-Theta running time of Swing. Briefly justify your answer. Recall that $\sum_{k=0}^n a^k = \frac{a^{n+1}-1}{a-1}$.

Solution: The number of leaves is $4^{\log_2 n} = 4^{\log_4 n \log_2 4} = n^{\log_2 4} = n^2$ which is $\Theta(n^2)$. The total number of nodes is proportional to the number of leaves (because $\sum_{k=0}^n 4^k = \frac{4^{n+1}-1}{3}$). Since each node contains a constant amount of work, the running time is proportional to the number of nodes. So the running time is $\Theta(n^2)$.

Name: _____

NetID: _____ Lecture: A B

Discussion: Thursday Friday 10 11 12 1 2 3 4 5 6

```

01 Wave( $a_1, \dots, a_n$ ) \\ input is an array of n positive integers
02      $m := 0$ 
03     for  $i := 1$  to  $n - 1$ 
04         for  $j := i + 1$  to  $n$ 
05             if  $|a_i - a_j| > m$  then  $m := |a_i - a_j|$ 
06     return  $m$ 

```

1. (3 points) What value does the algorithm return if the input list is 4, 13, 20, 5, 8, 10

Solution: $20 - 4 = 16$

2. (3 points) Let $T(n)$ be the number of times that line 5 is executed. Express $T(n)$ using summation notation, directly following the structure of the code.

$$\text{Solution: } T(n) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n 1$$

3. (3 points) Find an (exact) closed form for $T(n)$. Show your work.

$$\text{Solution: } T(n) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n 1 = \sum_{i=1}^{n-1} (n-i) = (\sum_{i=1}^{n-1} n) - (\sum_{i=1}^{n-1} i) = n(n-1) - \frac{n(n-1)}{2} = \frac{n(n-1)}{2}$$

4. (3 points) What is the big-theta running time of Wave?

Solution: $\Theta(n^2)$

5. (3 points) Check the (single) box that best characterizes each item.

The running time of mergesort is

recursively defined by $T(1) = d$ and

$T(n) =$

$2T(n-1) + c$ $2T(n-1) + cn$

$2T(n/2) + c$

$2T(n/2) + cn$