Name:_____

NetID:_____   **Lecture:**   A   B

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

```
01  Jump(a₁, ..., aₙ: an array of n positive integers)
02    if (n = 1) return a₁
03    else if (n = 2) return a₁ + a₂
04    else if (n = 3) return a₁ + a₂ + a₃
05    else
06        p = ⌊n/3⌋
07        q = ⌊2n/3⌋
08        rv = Jump(a₁, ..., aₚ) + Jump(a_{q+1}, ..., aₙ)
09        rv = rv + Jump(a_{p+1}, ..., a_q)
10        return rv
```

01 $\mathrm{Jump}(a_1, \ldots, a_n$: 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 $rv = \mathrm{Jump}(a_1, \ldots, a_p) + \mathrm{Jump}(a_{q+1}, \ldots, a_n)$
09 $rv = rv + \mathrm{Jump}(a_{p+1}, \ldots, a_q)$
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 $T(n)$.

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

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

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

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 $T(n)$.

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

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

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

Name:_____

NetID:_____ **Lecture:** **A** **B**

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

```
01  Waltz(a₁, a₂, ... aₙ: list of real numbers)
02    if (n = 1) then return 0
03    else if (n = 2) then return |a₁ − a₂|
04    else
05      L = Waltz(a₂,a₃,...,aₙ)
06      R = Waltz(a₁,a₂,...,aₙ₋₁)
07      Q = |a₁ − aₙ|
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.

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

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

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

Name:_____

NetID:_____   **Lecture:**   A   B

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

01 Grind$(a_1, \ldots, a_n)$ \\ 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$(a_1, \ldots, a_m)$ \\ constant time to extract part of array
07   else
08   return Grind$(a_{m+1}, \ldots, a_n)$ \\ 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 $T(n)$.

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

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

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

Name:——————————————————————————————

NetID:—————————————   **Lecture:**   **A**   **B**

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

```
01  Weave(a_0, ..., a_{n-1}))  \\ input is an array of n integers (n ≥ 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 = ⌊n/4⌋
06            q = ⌊n/2⌋
07            r = p + q
08            Weave(a_0, ..., a_q)   \\ constant time to make smaller array
09            Weave(a_{q+1}, ..., a_{n-1})   \\ constant time to make smaller array
10            Weave(a_p, ..., 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 $T(n)$.

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

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

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

Name:＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

NetID:＿＿＿＿＿＿＿＿＿＿＿＿＿   **Lecture:**   A   B

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

```
01  Act(a₁, ..., aₙ; b₁, ..., bₙ)  \\ input is 2 arrays of n integers, n is a power of 2
02        if (n = 1)
03              return a₁b₁
04        else
05              p = n/2
06              rv = Act(a₁, ..., aₚ, b₁, ..., bₚ)
07              rv = rv + Act(a₁, ..., aₚ, bₚ₊₁, ..., bₙ)
08              rv = rv + Act(aₚ₊₁, ..., aₙ, bₚ₊₁, ..., bₙ)
09              rv = rv + Act(aₚ₊₁, ..., aₙ, b₁, ..., bₚ)
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.

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

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

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

Name:

NetID: _____   **Lecture:**   A   B

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

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

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.

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

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

Name:_____

NetID:_____ **Lecture:** A B

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

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

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

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

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

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( (bottom+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 Jump(A,1,n) return?

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

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

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

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.

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

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

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

Name:_____

NetID:_____   **Lecture:**   A   B

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

```
01  Hoist(a₁,...,aₙ: an array of n positive integers)
02    if (n = 1) return 0
03    else if (n = 2) return a₁ + a₂)
04    else
05        p = ⌊n/3⌋
06        q = ⌊2n/3⌋
07        rv = max(Hoist(a₁,...,aₚ), Hoist(a_{q+1},...,aₙ))
08        for i=p to q
09            rv = max(rv, aᵢ + a_{i+1})
10        return rv
```

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

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

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

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 $\log_b x = \log_a x \log_b a$.

Name:_____

NetID:_____   **Lecture:**   **A**   **B**

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

```
01 Weave(a₁,..., aₙ)  \\ 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 aⱼ < a_min then min = j
06              swap(aᵢ,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?

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.

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

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

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(L₁,L₂: sorted lists of integers)
02        if (L₁ is empty)
03              return L₂
04        else if (L₂ is empty)
05              return L₁
06        else if (head(L₁) <= head(L₂))
07              return cons(head(L₁), Handle(rest(L₁),L₂))
08        else
09              return cons(head(L₂), Handle(L₁,rest(L₂)))
```

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 $T(n)$.

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

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

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

Name:_____

NetID:_____   **Lecture:**   **A**   **B**

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

```
01  Execute(p₁, ..., pₙ : list of n 2D points, n ≥ 3)
02       if (n = 3)
03            return the largest of d(p₁, p₂), d(p₁, p₃), and d(p₂, p₃)
04       else
05            x = Execute(p₂, p₃, p₄, ..., pₙ)      \\ removing p₁ from list takes constant time
06            y = Execute(p₁, p₃, p₄, ..., pₙ)      \\ removing p₂ from list takes constant time
07            z = Execute(p₁, p₂, p₄, ..., pₙ)      \\ removing p₃ 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 $T(n)$.

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

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

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

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 $T(n)$.

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

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

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

Name:_____

NetID:_____   **Lecture:**   A   B

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

```
01  Fabricate(a₁, …, aₙ; b₁, …, bₙ)  \\ input is 2 lists of n integers, n is a power of 2
02       if (n = 1)
03            return a₁b₁
04       else
05            p = n/2
06            rv = Fabricate(a₁, …, aₚ, b₁, …, bₚ)
07            rv = rv + Fabricate(a₁, …, aₚ, bₚ₊₁, …, bₙ)
08            rv = rv + Fabricate(aₚ₊₁, …, aₙ, bₚ₊₁, …, bₙ)
09            rv = rv + Fabricate(aₚ₊₁, …, aₙ, b₁, …, bₚ)
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.

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

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

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

Name:⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

NetID:⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯　　　**Lecture:**　　**A**　　**B**

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

```
01 Spin (a_1, ..., 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, ..., a_{n-1}) is true and Spin(a_2, ..., 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?

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.

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

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

Name:_____

NetID:_____　　　　**Lecture:**　　**A**　　**B**

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

```
01  Weave(a₁, ..., aₙ: an array of n positive integers)
02    if (n = 1) return 0
03    else if (n = 2) return max(a₁,a₂)
04    else
05       p = ⌊n/3⌋
06       q = ⌊2n/3⌋
07       rv = max(Weave(a₁, ..., aₚ), Weave(a_{q+1}, ..., aₙ))
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 $T(n)$.

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

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

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 $\log_b x = \log_a x \log_b a$.

Name:_____

NetID:_____   **Lecture:**   A   B

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

```
01  Knit(p₁, ..., pₙ : list of n 2D points, n ≥ 3)
02        if (n = 3)
03              return the largest of d(p₁, p₂), d(p₁, p₃), and d(p₂, p₃)
04        else
05              x = Knit(p₂, p₃, p₄, ..., pₙ)
06              y = Knit(p₁, p₃, p₄, ..., pₙ) \\ p₂ has been removed
07              z = Knit(p₁, p₂, ..., pₙ₋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.

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

2. (4 points) What is the amount of work (aka sum of the values in the nodes) at non-leaf level k of this tree?

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

4. (3 points) Is the running time of Knit $O(2^n)$?

Name:_____

NetID:_____   **Lecture:**   A   B

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

```
00 Churn(a₁, ..., aₙ) : list of n positive integers, n ≥ 2)
01     if (n = 2) return |a₁ - a₂|
02     else
03          bestval = 0
04          for k = 1 to n
05               newval = Churn(a₁, a₂, ..., aₖ₋₁, aₖ₊₁, ... aₙ)   \\ constant time to remove aₖ
06               if (newval > bestval) bestval = newval
07          return bestval
```

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

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 $T(n)$.

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

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

CS 173, Fall 17   Examlet 11, Part A

Name:_____

NetID:_____   **Lecture:**   A   B

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

```
01  Grind(a₁, a₂, . . . aₙ: list of real numbers)
02    if (n = 1) then return 0
03    else if (n = 2) then return |a₁ − a₂|
04    else
05      L = Grind(a₂,a₃,. . .,aₙ)
06      R = Grind(a₁,a₂,. . .,aₙ₋₁)
07      Q = |a₁ − aₙ|
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.

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

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

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

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.

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

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

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

Name:

NetID: **Lecture:**   **A**   **B**

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

```
01  Munch(a₁, . . . , aₙ: an array of n positive integers)
02    if (n = 1) return a₁
03    else if (n = 2) return a₁ + a₂
04    else if (n = 3) return a₁ + a₂ + a₃
05    else
06        p = ⌊n/3⌋
07        q = ⌊2n/3⌋
08        rv = Munch(a₁, . . . , a_p) + Munch(a_{q+1}, . . . , aₙ)
09        rv = rv + Munch(a_{p+1}, . . . , a_q)
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 $T(n)$.

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

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

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

Name:_____

NetID:_____   **Lecture:**   A   B

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

```
01  Crunch(a₀, . . . , aₙ₋₁))  \\ input is an array of n integers
02        if (n = 2 and a₀ > a₁)
03              swap(a₀, a₁)   \\ interchange the values at positions 0 and 1 in the array
04        else if (n > 2)
05                p = ⌊n/4⌋
06                q = ⌊n/2⌋
07                r = p + q
08                Crunch(a₀, . . . , a_q)   \\ constant time to make smaller array
09                Crunch(a_{q+1}, . . . , aₙ₋₁)   \\ constant time to make smaller array
10                Crunch(a_p, . . . , 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 $T(n)$.

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

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

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

Name:_____

NetID:_____   **Lecture:**   A   B

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

```
01  Crumple(a₁, ..., aₙ: a list of n positive integers)
02    if (n = 1) return a₁
03    else if (n = 2) return a₁ + a₂
04    else if (n = 3) return a₁ + a₂ + a₃
05    else
06        p = ⌊n/3⌋
07        q = ⌊2n/3⌋
08        rv = Crumple(a₁, ..., aₚ) + Crumple(a_{q+1}, ..., aₙ)
09        rv = rv + Crumple(a_{p+1}, ..., 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 $T(n)$.

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

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

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

Name:_____

NetID:_____   **Lecture:**   A   B

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

```
01  Slide(a₁, ..., aₙ)  \\ input is a linked list of n integers
02        if (n = 1) return a₁
03        else
04                m = ⌊n/2⌋
05                p = Slide(a₁, ..., aₘ)      \\ O(n) time to split list
06                q = Slide(aₘ₊₁, ..., aₙ)    \\ 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 $T(n)$.

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

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?

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

Name:

NetID: **Lecture:**   **A**   **B**

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

```
01  Swing(a₁, ..., aₙ; b₁, ..., bₙ)  \\ input is 2 arrays of n integers, n is a power of 2
02        if (n = 1)
03                return a₁b₁
04        else
05                p = n/2
06                rv = Swing(a₁, ..., aₚ, b₁, ..., bₚ)
07                rv = rv + Swing(a₁, ..., aₚ, bₚ₊₁, ..., bₙ)
08                rv = rv + Swing(aₚ₊₁, ..., aₙ, bₚ₊₁, ..., bₙ)
09                rv = rv + Swing(aₚ₊₁, ..., aₙ, b₁, ..., bₚ)
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.

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

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

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

Name:_____

NetID:_____   **Lecture:**   A   B

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

```
01  Wave(a₁,..., aₙ)  \\ 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 |aᵢ − aⱼ| > m then m := |aᵢ − aⱼ|
06      return m
```

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

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.

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

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

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 $T(n) =$

$2T(n − 1) + c$ ☐ $2T(n − 1) + cn$ ☐

$2T(n/2) + c$ ☐ $2T(n/2) + cn$ ☐