CS 173, Spring 19 Examlet 11, colored sheet

Name:____

NetID:______________________ **Lecture: A B**

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

```
Jump(a_1, ..., a_n: an array of n positive integers)
  if (n = 1) return a_1
  else if (n = 2) return a_1 + a_2
  else if (n = 3) return a_1 + a_2 + a_3
  else
      p = ⌊n/3⌋
      q = ⌊2n/3⌋
      rv = Jump(a_1, ..., a_p) + Jump(a_{q+1}, ..., a_n)
      rv = rv + Jump(a_{p+1}, ..., a_q)
      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**

```
Swing(k,n)   \\ inputs are positive integers
    if (n = 1) return k
    else if (n = 2) return k^2
    else
        half = ⌊n/2⌋
        answer = Swing(k,half)
        answer = answer*answer
        if (n is odd)
            answer = answer*k
        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?

CS 173, Spring 19 Examlet 11, colored sheet

Name:____

NetID:____________________ **Lecture: A B**

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

```
Waltz(a_1, a_2, ... a_n: list of real numbers)
  if (n = 1) then return 0
  else if (n = 2) then return |a_1 − a_2|
  else
    L = Waltz(a_2,a_3,...,a_n)
    R = Waltz(a_1,a_2,...,a_{n−1})
    Q = |a_1 − a_n|
    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

```
Grind(a_1, ..., a_n)  \\ input is a sorted array of n integers
        if (n = 1) return a_1
        else
               m = ⌊n/2⌋
               if a_m > 0
                      return Grind(a_1, ..., a_m)   \\ constant time to extract part of array
               else
                      return Grind(a_{m+1}, ..., 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

```
Weave(a_0, ..., a_{n-1}))  \\ input is an array of n integers (n ≥ 2)
        if (n = 2 and a_0 > a_1)
                swap(a_0, a_1)   \\ interchange the values at positions 0 and 1 in the array
        else if (n > 2)
                p = ⌊n/4⌋
                q = ⌊n/2⌋
                r = p + q
                Weave(a_0, ..., a_q)   \\ constant time to make smaller array
                Weave(a_{q+1}, ..., a_{n-1})   \\ constant time to make smaller array
                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

```
Act(a_1, ..., a_n; b_1, ..., b_n)  \\ input is 2 arrays of n integers, n is a power of 2
    if (n = 1)
        return a_1 b_1
    else
        p = n/2
        rv = Act(a_1, ..., a_p, b_1, ..., b_p)
        rv = rv + Act(a_1, ..., a_p, b_{p+1}, ..., b_n)
        rv = rv + Act(a_{p+1}, ..., a_n, b_{p+1}, ..., b_n)
        rv = rv + Act(a_{p+1}, ..., a_n, b_1, ..., b_p)
        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 $\sum_{k=0}^{n} a^k = \frac{a^{n+1}-1}{a-1}$.

CS 173, Spring 19 Examlet 11, colored sheet

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 (Dig($a_1, \ldots, a_{n-1}$) is true and 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

```
Swim(a_1, ..., a_n)  \\ input is a sorted list of n integers
      if (n = 1) return a_1
      else
            m = ⌊n/2⌋
            if a_m > 0
                  return Swim(a_1, ..., a_m)   \\ O(n) time to extract half of list
            else
                  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?