CS 173, Spring 18 Examlet 11, Part A

Name:_____

NetID:_____ **Lecture: A B**

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

```
01 Hoist(a_1, ..., 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 = ⌊n/3⌋
06       q = ⌊2n/3⌋
07       rv = max(Hoist(a_1, ..., a_p), Hoist(a_{q+1}, ..., 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 $T(n)$.

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(a_1,..., 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?

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(p_1, ..., p_n : list of n 2D points, n ≥ 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, ..., p_n)     \\ removing p_1 from list takes constant time
06            y = Execute(p_1, p_3, p_4, ..., p_n)     \\ removing p_2 from list takes constant time
07            z = Execute(p_1, p_2, p_4, ..., 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 $T(n)$.

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(a_1, ..., a_n; b_1, ..., 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 = n/2
06             rv = Fabricate(a_1, ..., a_p, b_1, ..., b_p)
07             rv = rv + Fabricate(a_1, ..., a_p, b_{p+1}, ..., b_n)
08             rv = rv + Fabricate(a_{p+1}, ..., a_n, b_{p+1}, ..., b_n)
09             rv = rv + Fabricate(a_{p+1}, ..., a_n, b_1, ..., 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.

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.)