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

```
Hoist(a_1, ..., a_n: an array of n positive integers)
 if (n = 1) return 0
 else if (n = 2) return a_1 + a_2)
 else
     p = ⌊n/3⌋
     q = ⌊2n/3⌋
     rv = max(Hoist(a_1, ..., a_p), Hoist(a_{q+1}, ..., a_n))
     for i=p to q
         rv = max(rv, a_i + a_{i+1})
     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: $2^{\log_3 n} = n^{log_3 2}$

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 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?

   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 ith 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

```
Handle(L1,L2: sorted lists of integers)
    if (L1 is empty)
        return L2
    else if (L2 is empty)
        return L1
    else if (head(L1) <= head(L2))
        return cons(head(L1), Handle(rest(L1),L2))
    else
        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 $T(n)$.

   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.)

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**

```
Execute(p_1, ..., p_n : list of n 2D points, n ≥ 3)
    if (n = 3)
        return the largest of d(p_1, p_2), d(p_1, p_3), and d(p_2, p_3)
    else
        x = Execute(p_2, p_3, p_4, ..., p_n)    \\ removing p_1 from list takes constant time
        y = Execute(p_1, p_3, p_4, ..., p_n)    \\ removing p_2 from list takes constant time
        z = Execute(p_1, p_2, p_4, ..., p_n)    \\ removing p_3 from list takes constant time
        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: $\Theta(3^n)$

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**

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

```
Fabricate(a_1, ..., a_n; b_1, ..., b_n)  \\ input is 2 lists of n integers, n is a power of 2
        if (n = 1)
                return a_1 b_1
        else
                p = n/2
                rv = Fabricate(a_1, ..., a_p, b_1, ..., b_p)
                rv = rv + Fabricate(a_1, ..., a_p, b_{p+1}, ..., b_n)
                rv = rv + Fabricate(a_{p+1}, ..., a_n, b_{p+1}, ..., b_n)
                rv = rv + Fabricate(a_{p+1}, ..., a_n, b_1, ..., b_p)
                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: $4^{\log_2 n} = 4^{\log_4 n \log_2 4} = n^{\log_2 4} = n^2$