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 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?

   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**

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

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**

Knit($p_1, \ldots, p_n$: list of n 2D points, $n \geq 3$)
    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 = Knit($p_2, p_3, p_4, \ldots, p_n$)
        y = Knit($p_1, p_3, p_4, \ldots, p_n$) \\\\ p_2 has been removed
        z = Knit($p_1, p_2, \ldots, p_{n-1}$)
        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 $T(n)$.

   Solution: $T(3) = c$

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

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?

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

   Solution: 3^{n-3}

4. (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

```
Churn(a_1, ..., a_n) : list of n positive integers, n ≥ 2)
    if (n = 2) return |a_1 − a_2|
    else
        bestval = 0
        for k = 1 to n
            newval = Churn(a_1, a_2, ..., a_{k−1}, a_{k+1}, ... a_n)   \\ constant time to remove a_k
            if (newval > bestval) bestval = newval
        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: $\frac{n!}{2}$

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

```
Grind(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 = Grind(a_2,a_3,...,a_n)
    R = Grind(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 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}

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

```
Sew(k,n)   \\ inputs are natural numbers
        if (n = 0) return 1
        else if (n = 1) return k
        else if (n is odd)
              temp = Sew(k,floor(n/2))
              return k*temp*temp
        else
              temp = Sew(k,floor(n/2))
              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: $\Theta(\log n)$