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?

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(a_1, ..., a_n: an array of n positive integers)
02   if (n = 1) return 0
03   else if (n = 2) return max(a_1,a_2)
04   else
05       p = ⌊n/3⌋
06       q = ⌊2n/3⌋
07       rv = max(Weave(a_1, ..., a_p), Weave(a_{q+1}, ..., a_n))
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(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 = Knit(p_2, p_3, p_4, ..., p_n)
06         y = Knit(p_1, p_3, p_4, ..., p_n) \\ p_2 has been removed
07         z = Knit(p_1, p_2, ..., p_{n-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(a_1, ..., a_n) : list of n positive integers, n ≥ 2)
01      if (n = 2) return |a_1 − a_2|
02      else
03          bestval = 0
04          for k = 1 to n
05              newval = Churn(a_1, a_2, ..., a_{k−1}, a_{k+1}, ... a_n)   \\ constant time to remove a_k
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 $T(n)$?

Name:_____

NetID:_____ **Lecture: A B**

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

```
01 Grind(a_1, a_2, ... a_n: list of real numbers)
02   if (n = 1) then return 0
03   else if (n = 2) then return |a_1 − a_2|
04   else
05     L = Grind(a_2,a_3,...,a_n)
06     R = Grind(a_1,a_2,...,a_{n−1})
07     Q = |a_1 − a_n|
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?