CS 325 - Activity 2
You may work in groups with up to 3 students. When submitting solutions in Gradescope select a page for each problem and the students in your group.

Problem 1: (5pts) Consider the following algorithm:

```
def Goo(n):
  if n <= 1:
    return 1
  else:
    x = Goo(n-2)
    sum = x + x
    return sum
```

(a) Write a recurrence for the running time T(n) of **Go**o(n).

(b) Solve the recurrence for the asymptotic running time. Assume that addition can be done in constant time. Use theta notation.

CS 325 - Activity 2
You may work in groups with up to 3 students. When submitting solutions in Gradescope select a page for each problem and the students in your group.

Problem 2: (5pts) Consider the following algorithm:

```
Foo(n) {
    total = 0
    if n = 1 return 2
    else {
        total = Foo(n/4) + Foo(n/4)
         for i = 1 to n do
             for j = 1 to n do
                 total = total + i*j
        return total
     }
}
```

(a) Write a recurrence for the running time T(n) of **Foo**(n).

(b) Solve the recurrence for the asymptotic running time. Assume that addition can be done in constant time. Use theta notation.

CS 325 - Activity 2

You may work in groups with up to 3 students. When submitting solutions in Gradescope select a page for each problem and the students in your group.

Problem 3: (5 pts) Consider the following cruel and unusual sorting algorithm.

```
CRUEL(A[1 .. n]):
  if n > 1
    Cruel(A[1, .., n/2])
    Cruel(A[n/2 + 1, .., n])
    Unusual(A[1, .. ,n])
```

```
UNUSUAL(A[1 .. n]):
  if n = 2                          //the only comparison!
    if A[1] > A[2]
      swap a[1] with A[2]
  else
    for i from 1 to n/4             //swap 2nd and 3rd quarters
      swap A[i + n/4] with A[i + n/2]
    Unusual(A[1, .. ,n/2])                //recurse on left half
    Unusual(A[n/2 + 1, .. ,n])            //recurse on right half
    Unusual(A[n/4 + 1, .. ,3n/4])         //recurse on middle half
```

(a) Give a recurrence for the running time T(n) of the **UNUSUAL** function.

(b) Solve the recurrence in part a) to find the running time of **UNUSUAL** use Theta notation.

EXTRA Credit. Give a recurrence for **CRUEL** and solve it.