

CS30 Spring 2015
Extra Credit
Due: Friday, June 12, 8:00 am

Submit code for the functions you write. Submit by emailing shinar@cs.ucr.edu.

1. (40 points) Maximum sum sublist. Given a list (1D array) of integers, both positive and negative, write a function `MaximumSumSublist` which finds the contiguous sublist with the largest possible sum. Your function should return the sum of the numbers in the sublist, as well as the start and end indices of the sublist. I.e.,

```
function [sum, startIndex, endIndex] = MaximumSumSublist(list)
```

For example, the box shows the sublist giving the maximum sum and the result of calling `MaximumSumSublist`.

[2, -5, 4, 3, -1, 2, -5, 4]

```
>> [sum, startIndex, endIndex] = MaximumSumSublist(list)
sum =
     8
startIndex =
     3
endIndex =
     6
```

There are several different ways to solve this problem, with varying degrees of efficiency. How many times does your code visit each element in the list? Make a plot of list size vs. number of list accesses to see how efficient your code is. Try various lists of increasing length until you see the trend in length/cost or the computation starts taking more than a minute or two. Submit your plot along with your code.

2. (60 points) Animated bouncing circle. Use Matlab to animate the bouncing of a circle in a box. Your function `BounceCircle` should take as input the initial position and velocity of the circle, a coefficient of restitution, $C \in [0, 1]$, and an array with the extents of the box [`xmin xmax ymin ymax`]. This is similar to the `Projectile` example we looked at in class and in the reading, and animated in class. However, in this case, you should detect when your circle hits a wall, and it should bounce off reflecting about the normal. The speed at which it bounces off will $C * v$, where v is the speed at which it hits the wall. Also, you do not need to include gravity. Submit your function code.