

## Class 10 - Notes

### Upcoming Schedule

[Project 2](#) is due at the beginning of class on **Monday, 15 February**.

Yuchi has office hours right **after class today**, and has scheduled additional office hours, **3-4:30pm on Sunday in Rice 532**.

### Gravitational Waves

P. B. Abbot, et al., *Observation of Gravitational Waves from a Binary Black Hole Merger*. Physical Review Letters, 12 February 2016.

The Guardian, *Gravitational waves: breakthrough discovery after two centuries of expectation*, 11 February 2016. (Best popular explanation of what they did with figures.)

[Git Repositories for LIGO project signals.py](#)

### Programming with Lists

```
def make_pair(a, b):  
    return [a, b]
```

```
def pair_first(pair):  
    return pair[0]
```

```
def pair_last(pair):  
    return pair[1]
```

```
# A list is a pair where the second part is a list, or None
```

```
def make_list(first, second):  
    return make_pair(first, second)
```

```
def list_first(lst):  
    return pair_first(lst)
```

```
def list_rest(lst):  
    return pair_last(lst)
```

**Defining List Procedures:**

1. *Be very optimistic!* Since lists themselves are recursive data structures, most problems involving lists can be solved with recursive procedures.
2. **Base Case.** Think of the simplest version of the problem, something you can already solve. This is the base case. For lists, this is usually when the list is empty or a singleton.
3. **Recursive Case.** Consider how you would solve the problem using the result for a slightly smaller version of the problem. This is the recursive case. For lists, the smaller version of the problem is usually the rest of the list. So, the procedure combines the result of doing something with the first element ( $p[0]$ ) with making a recursive call on the rest of the elements ( $p[1:]$ ).

```
def list_length(lst):  
    # Returns the number of elements in lst.
```

```
def list_flip(lst):  
    # Returns a new list, where each element  
    # is the negation of the elements in the input lst.
```