

# Problem Wk.7.2.5: Argopt

---

## Part 1: Argmin

Define a procedure `argmin` that takes two arguments, a procedure of one argument and a list. It returns a tuple `(bestValue, bestArg)` with the min value of the input function over the input list and the value in the list that gives that min value.

```
>>> argmin(lambda x: (x-3)**2, [1,2,3,4])
(0, 3)
```

`lambda` is a way of creating a simple function without giving it a name; you should read about it in the course readings (Chapter 3.4.6). The example above is exactly equivalent to the example below, where we have created the function in the usual way. Note that `f` evaluates to the function itself.

```
>>> def f(x): return (x-3)**2
>>> argmin(f, [1,2,3,4])
(0, 3)
```

```
def argmin(f, input):
    bestValSoFar = None
    bestArgSoFar = None
    for x in input:
```

---

## Part 2: Argopt

Define a procedure `argopt` that takes three arguments, a procedure of one argument, a list, and a comparison function. The comparison function, for example, could be less than or greater than, but we won't assume that it is only that, it could be something else.

`argopt` returns a tuple `(bestVal, bestArg)` with the 'optimal' value using the comparison function and the entry in the list that gives that value.

In Python, the comparison operators, just like the arithmetic operators, are not standard functions, so you can not pass them to other functions. However if one does `import operator` that defines procedures that do the job of those operators:

- `operator.lt` is '<'
- `operator.gt` is '>'
- `operator.add` is '+'
- `operator.sub` is '-'

for example:

```
>>> operator.lt(2,4)
True
>>> operator.gt(2,4)
False
```

So, this would be equivalent to `argmin`

```
>>> argopt(lambda x: (x-3)**2, [1,2,3,4], operator.lt)
(0, 3)
>>> argopt(lambda x: x[0], [(1,2),(3,4)], operator.lt)
(1, (1, 2))
```

Hint: this is a tiny change from `argmin`

```
def argopt(f, input, comp):
    bestValSoFar = None
    bestArgSoFar = None
    for x in input:
```

MIT OpenCourseWare  
<http://ocw.mit.edu>

6.01SC Introduction to Electrical Engineering and Computer Science  
Spring 2011

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.