

In class we gave the following sorting code.

```
qsort [] = []
qsort (h:hs) = qsort smaller ++ [h] ++ qsort larger
  where
    smaller = [a | a <- hs, a < h]
    larger  = [b | b <- hs, b >= h]
```

The notation `[ b | b <- hs, b >= h ]` is called a list comprehension. As described in class, the expression `b <- hs` says, iterate over the elements of the list `hs` and collect together into a list all the `b`'s where `b >= h`.

Once loaded into HUGS, checking the type gives the following result.

```
> :t qsort
qsort :: Ord a => [a] -> [a]
```

I suggested the following change to modify the type.

```
qsort [] = []
qsort' (h:hs) = qsort' smaller ++ [h] ++ qsort' larger
  where
    smaller = [a + 1 | a <- hs, a < h]
    larger  = [b + 1 | b <- hs, b >= h]
```

In fact, it does have the type that I suggested.

```
> :t qsort'
qsort' :: (Ord a, Num a) => [a] -> [a]
```

But this program does not have the behavior I suggested. Here is the behavior it has.

```
Main> qsort' (reverse [1..10])
qsort' (reverse [1..10])
[10,10,10,10,10,10,10,10,10,10]
Main> qsort' [1..10]
qsort' [1..10]
[1,3,5,7,9,11,13,15,17,19]
Main> qsort' [1,10,9,2,8,3,7,4,6,5]
qsort' [1,10,9,2,8,3,7,4,6,5]
[1,5,8,11,14,14,13,12,11,11]
Main> qsort' [10,1,9,2,8,3,7,4,6,5]
[2,5,8,11,14,14,13,12,11,10]
Main>
```

What's happening here? How can the program be fixed to give the answer? Here's test cases and expected results for the program I thought I was writing.

```

Main>:t qsort1
qsort1 :: (Num a, Ord a) => [a] -> [a]
Main> qsort1 (reverse [1..10])
qsort1 (reverse [1..10])
[2,3,4,5,6,7,8,9,10,11]
Main> qsort1 [1..10]
qsort1 [1..10]
[2,3,4,5,6,7,8,9,10,11]
Main> qsort1 [1,10,9,2,8,3,7,4,6,5]
qsort1 [1,10,9,2,8,3,7,4,6,5]
[2,3,4,5,6,7,8,9,10,11]
Main> qsort1 [10,1,9,2,8,3,7,4,6,5]
qsort1 [10,1,9,2,8,3,7,4,6,5]
[2,3,4,5,6,7,8,9,10,11]

```

Your assignment is to:

- 1.) Explain the behavior of `qsort'`.
- 2.) Modify `qsort'` creating a function `qsort1` that behaves properly.
- 3.) To run the tests to show your function behaves as expected.