

UMassAmherst

Manning College of Information
& Computer Sciences

Programming Methodology

Lab 3

Wednesday September 20, 2023



Weekly Lab Agenda

- Go over reminders/goals
- Review past material
- Work in groups of 2-3 to solve a few exercises
 - Please sit with your group from last week.
- Discussion leaders will walk around and answer questions
- Solutions to exercises will be reviewed as a class
- Attendance taken at the end

Reminders

- Homework 2 is due tonight at 11:59pm
 - Come to office hours for help!
- Homework 3 releases soon.
- Start reviewing for midterm 1!

Today's Goals

- Lists
- Reduce

Lists are **recursive**

Lists are either

- empty
- or an **element** followed by a **list**

Lists are an **abstract data type** with the following methods:

- **list.isEmpty()**
 - returns **True** if the list is empty()
 - returns **False** if the list is **node(data,next)** [an element followed by a list]
- **list.head()**
- **list.tail()**

Exercise 1

Given 2 ordered lists, merge them such that resulting list is an ordered list.

```
// merges two ordered lists  
function merge(list1: List<number>, list2: List<number>): List<number>
```

Exercise 1 - Solution

```
// merges two ordered lists

function merge(l1: List<number>, l2: List<number>) {
  if (l1.isEmpty()) return l2;
  if (l2.isEmpty()) return l1;
  const h1 = l1.head();
  const h2 = l2.head();
  return h1 < h2
    ? node(h1, merge(l1.tail(), l2))
    : node(h2, merge(l1, l2.tail()));
}
```

Review of Reduce

```
function reduce<T, U>(
  a: T[],
  f: (acc: U, e: T) => U,
  init: U
): U {
  let result = init;
  for (let i = 0; i < a.length; ++i) {
    result = f(result, a[i]);
  }
  return result;
}
```

Reduce is used to combine array elements with the same function.

Example: Find the product of all elements of an array `a = [3, 2, 6, 2, 2, 0]`

```
a.reduce((prod, e) => prod * e, 1);
```


Exercise 2

Return the sum of all positive and the sum of all negative numbers from an array.

```
function sumPositivesAndNegatives(arr: number[]): [number, number]
```

Exercise 2 - Solution

```
function sumPositivesAndNegatives(arr: number[]): [number, number] {  
  return arr.reduce(  
    ([positive, negative], curr) =>           // no effect if curr is 0  
    (curr > 0) ? [positive + curr, negative] : [positive, negative + curr],  
    [0, 0]  
  );  
}
```