COMP161 - Lab 9 and Homework 6

Spring 2014

In this lab you'll tackle iterative and recursive procedures for vectors.

Choose your own adventure

You should tackle these problems one at a time in any order you want[1]. Each problem can and should be done using iterative loops and recursion[2]. In some cases, there are functional and stateful versions you should do as well. As a reminder, functional procedures use function input and output without any side-effects. This means no I/O and no mutation. Stateful procedures are typically variable mutators. In some cases you may need to mutate the vector itself.

1. *isSorted*

   This predicate takes a vector of integers and returns true if the contents of the vector are in least to greatest sorted order and false otherwise. This problem should be solved functionally using iteration and recursion.

2. *find* a.k.a. *search*

   Given a vector of integers and a target number called the *key*, find and return the location of the first occurrence of the key or if the key is not in the vector return -1. So, if a vector contains $\{0, 2, 4, 6, 8\}$ and the key is 4, you should return 2. If for that same vector the target was 1, you'd return -1. This problem should be solved functionally using iteration and recursion.

3. *setToOdds*

   This is the stateful version of *mapToOdds*. Using *setToOdds* modifies the existing vector rather than returning a new vector. Using *setToOdds* on a vector containing $\{2, 4, 6, 8\}$ should change that vector's contents to $\{5, 9, 13, 17\}$. This problem should be solved statefully using iteration and recursion.

4. *mapToOdds*

   The function *setToOdds* takes a vector of integers and computes the vector of all the odds for the given content. If the original vector contains the number 4, this is mapped to the 4th odd number, $2 * 4 + 1 = 9$. So, the using *mapToOdds* on a vector containing $\{2, 4, 6, 8\}$ should return the vector containing $\{5, 9, 13, 17\}$. This problem should be solved functionally using iteration and recursion.

[1] Note that they are listed in order of difficulty

[2] See Lecture notes 16 for more guidance

5. *evensFilter*

   This functional procedure should, when given a vector of integers, return the vector containing all the evens in the original vector. So, given a vector containing $\{1,2,3,4,5,6\}$, *evensFilter* should return a vector containing $\{2,4,6\}$. For this problem you'll need to dynamically size the vector. This will require some vector class methods we haven't seen so be ready to explore the documentation[3]. Here's a hint: either start with the max possible size you'd need, then resize after you've filled it with evens, or add and increase the size as you go. This problem should be solved functionally using recursion and iterative loops.

6. *removeOdds*

   This is the stateful version of *evensFilter*. Given a vector containing $\{1,2,3,4,5,6\}$, *removeOdds* should modify the contents such that they're now the vector $\{2,4,6\}$. Just as was the case with the functional version, this will require resize a vector. This problem should be solved statefully using recursion and iterative loops.

Lab 9 Assignment

Begin work on the problems and when lab is over submit what you have as *lab9* using handin. Remember, do the problems one at a time, not as a batch. I don't want to see declarations and tests for everything with no implementation. I'd rather see one of the problems fully done and nothing for the others.

Homework 6 Assignment

Complete the problem set and submit the complete set *before the next lab period*. Submissions should be done via handin. The assignment is, of course, *hwk6*.

[3] http://www.cplusplus.com/reference/vector/vector/