

CISC4080

Fall 2022

Self-test Exercises #1

Recall the four “patterns” of processing a list/array of items, and solve problem 2 and 3 using such patterns:

- Scan all elements from beginning to end or the from end to beginning
- Using two indices: one starting from beginning and one starting from the end, and make them merge in the middle
- process **all pairs of adjacent elements** in the list: as in bubble sort
 - $a[1]$ with $a[2]$, $a[2]$ with $a[3]$, ... , $a[i]$ with $a[i+1]$, ..., $a[n-1]$ with $a[n]$
- process **all possible pair of elements** in the list: as in checking if a list contains duplicate values or not
 - This is usually done by the following order
 - $a[1]$ with $a[2]$, $a[1]$ with $a[3]$, $a[1]$ with $a[4]$, ... $a[1]$ with $a[n]$
 - $a[2]$ with $a[3]$, $a[2]$ with $a[4]$, $a[2]$ with $a[5]$, ... $a[2]$ with $a[n]$
 - ...
 - $a[n-1]$ with $a[n]$

in code:

```
for  $i=1$  to  $n-1$ 
    for  $j=i+1$  to  $n$ 
        do something with  $a[i]$  and  $a[j]$ , such as comparison or calculate
        the difference
```

- process **pairs of elements from both ends of the list:**
first element with last, second element with second last, and so on

Example: check if a string is palindrome

in code: //i, j starts from two ends of list, moving towards to the middle until they meet:

```
for ( $i=1, j=n; i < j; i++, j--$ )
    do something with  $a[i]$  and  $a[j]$ 
```

1. Write pseudocode to reverse a list

```
/* Reverse list a
  @param a: the list to be reversed
  @param n: length of list a
  @postcondition: list a is reversed
*/
ReverseList (a, n)
```

Or write a C++ function to reverse a vector of int

```
/* Reverse vector a
  @param a: the vector of int to be reversed
  @postcondition: vector a is reversed */
void ReverseVector(vector<int> & a)
{
    int n = a.size();

}
}
```

2. *Write pseudocode or C++ code to check if a string is a Palindrome*
3. *Write pseudocode or C++ code to check if a list of int contains two numbers that add up to 100 or not*
4. *Write pseudocode or C++ code to check if a list of int contains -1 or not*
5. *You should be able to write/read/reason/trace through Bubble Sort*
6. *You should be able to write/read/trace/reason/trace through Selection Sort*