UCR - CS 010C

Assignment 1

Deliverables: Create a single pdf file that contains your answers and your C++ code. Then create a zip file that contains this pdf file along with all your code source files. Submit this zip file in iLearn.

Deadline: 10/16/2020 11:59 pm.

Exercise 1:

A. Using only core C++ (no special libraries, except STL vector or string if you want), write a C++ program that allows a user to input a string and

(a) Checks if the expression is a valid polynomial. Parentheses or negation are not allowed. Spaces should be ignored. E.g., the following are valid
   i. *n^2+2*n+5*
   ii. *2*n + 4.54* n^5 +4 +5*n*
   and the following are invalid
   iii. *n^3n*
   iv. *n^4.2*
   v. *5n*
   vi. *n^3 -3*n*

(b) If the polynomial is valid, outputs its big-O notation. E.g., for (ii) above it is *O(n^5)*.

B. If the length of the input expression is *m* chars, what is the big-O complexity of your program with respect to *m*?

C. What if we require that there is only one term for each degree? That is, (ii) above is invalid because it has two terms for degree *1* (*n^1*).

Modify your program accordingly.

What is the asymptotic complexity of the new program?

Throughout the exercise, make any assumptions necessary.

Exercise 2:

Given an array *A* of *n* integers and an integer *s*, find a subset of the integers in *A* such that their product is *s*.

A. Write C++ function.

B. Compute asymptotic complexity.

Exercise 1

```
#include <iostream>
#include <vector>
#include <string>

using namespace std;

void count (string in) {

}

int main() {
   // string temp = "";
   string polyIn = "";
   bool polynomial = false;
   string bigO = "";

   cout << "Input a polynomial:" <<  endl;
   getline(cin, polyIn);

   for (int i = 0; i < polyIn.size(); i++) {
      if (polyIn[i] == '(') {
         cout << "The polynomial is invalid." << endl;
         polynomial = false;
      } else if (polyIn[i] == ')') {
```

```
        cout << "The polynomial is invalid." << endl;
        polynomial = false;
    }

    if (polyIn[i] == ' ') {
        polyIn.erase(remove(polyIn.begin(), polyIn.end(), ' '), polyIn.end());
        // temp = polyIn;
        // cout << polyIn << endl; // debug for remove spaces.
        polynomial = true;
    }

    if (polyIn[i] == 'n') {
        if (polyIn[i+1] == '^') {
            bigO = polyIn[i] + polyIn[i+1] + polyIn [i+2];
        } else {
            bigO = polyIn[i];
        }
    }

}

if (polynomial == true) {
        cout << "O(" << bigO << ")" << endl;
    }
```

```
}
```