    1 int m = 5, n = 7; // m =5, n = 7

    2 int \*p; // m = 5, n = 7

    3 p = &n; // \*p = n = 7, m = 5

    4 \*p = 2 \* n + m; // \*p = 2 \* 7 + 5 = 19 = n, m = 5

    5 \*p = 2 \* n + m; // \*p = 2 \* 19 + 5 = 43 = n, m = 5

    6 p = &m; // \*p = m = 5, n = 43

    7 \*p = 2 \* m + n; // \*p = 2 \* 5 + 43 = 53 = m, n = 43

    8 \*p = 2 \* m + n; // \*p = 2 \* 53 + 43 = 149 = m, n = 43

    9 cout << "m = " << m << ", "; // m = 149

   10 cout << "n = " << n << ", "; // n = 43

   11 cout << "\*p = " << \*p << endl; // \*p = 149

int x; // 3173315904

int \*ptr = &x; // &ptr = &x = 3173315904

ptr++; // &ptr = 3173315905

The graph is not properly initialized

    int arr[5] = {1,2,3,4,5} ;

    int \*ptr = arr ;

    cout << \*(ptr + 2) << endl ;

Output:

    3

    5

leo.Eats();

lionPtr->Sleep();

It’s a variable in a class

Leo eats meat.

waBegin and waEnd return iterators pointing to the first and last elements of the array

WideArray::waIterator::operator++() {

        m\_ptr+=2;

}

WideArray::waIterator waEnd() {

    return waIterator(m\_data+2\*m\_size)

}

Segmentation fault because there isn’t a declared variable at this address

The old data array is not deallocated. I would use valgrind to detect it.