CS 477 Extra Difficulty Assignment

Mergesort a Linked List Using Constant Workspace

It is usual when coding mergesort to copy half the original list into another list. This method uses $\Theta(n)$ *workspace*, defined to be the space requirement of the program over and above that required to store the input data. The project is to write C++ code for mergesort of a linked list which uses $O(1)$ workspace.

Below is my code. I define a linked list type. I generate list with randomly chosen integral values, then sort that list with the function `mergesort`. My program uses `new` only when constructing my input list, in the funtion `randomlist`. Alternatively, I could have written the program to read a list from a file. There are two missing functions in my program, `split`, which splits the original list into two lists, and and `merge`, which merges two sorted lists. Each function in the program, other than `randomlist`, uses $O(1)$ workspace.

```
//Mergesort a linked list
//
 #include<iostream>
 #include<cassert>
 #include<cmath>
 #include<string>
 #include<cstdio>
 #include<iomanip>
 #include <stdio.h>      /* printf, scanf, puts, NULL */
 #include <stdlib.h>     /* srand, rand */
 #include <time.h>
 #include<fstream>
 #include <cstdlib>
  using namespace std;

  const int N = 20;

  struct list
   {
    int item;
    list*link;
   };

  void writelist(list*ell)
   {
    if(ell)
     {
      cout << ell->item << " " << endl;
      writelist(ell->link);
     }
   }

  list*randomlist()
   {
    list*rslt = NULL;
    for(int i = 0; i < N; i++)
     {
      list*ell = new list;
      ell->item = rand();
      ell->link = rslt;
      rslt = ell;
     }
    return rslt;
   }
```

Functions `split` and `merge` deleted.

```
 void mergesort(list*&ell)
  {
   if(ell)
    if(ell->link)
     {
      list*ell1 = NULL;
      list*ell2 = NULL;
      split(ell,ell1,ell2);
      mergesort(ell1);
      mergesort(ell2);
      ell = merge(ell1,ell2);
     }
  }

void mainwork()
 {
  srand(1);
  list*newlist = randomlist();
  writelist(newlist);
  mergesort(newlist);
  cout << endl << "Sorted list" << endl << endl;;
  writelist(newlist);
 }

int main()
 {
  mainwork();
  return 1;
 }
```

Output of my Program:

```
1303455736
304089172
1540383426
1365180540
1967513926
2044897763
1102520059
783368690
1350490027
1025202362
1189641421
596516649
1649760492
719885386
424238335
1957747793
1714636915
1681692777
846930886
1804289383

Sorted list

304089172
424238335
596516649
719885386
783368690
846930886
1025202362
1102520059
1189641421
1303455736
1350490027
1365180540
1540383426
1649760492
1681692777
1714636915
1804289383
1957747793
1967513926
2044897763
```