Term paper

Question\_1:

merge\_sort(["date", "cherry", "apple", "banana"], 0, 3)

/ \

merge\_sort(["date", "cherry"], 0, 1) merge\_sort(["apple", "banana"], 2, 3)

/ \ / \

merge\_sort(["date"], 0, 0) merge\_sort(["cherry"], 1, 1) merge\_sort(["apple"], 2, 2) merge\_sort(["banana"], 3, 3)

| | | |

["date"] ["cherry"] ["apple"] ["banana"]

\ / \ /

merge(["date", "cherry"]) merge(["apple", "banana"])

| |

["cherry", "date"] ["apple", "banana"]

\ /

merge(["apple", "banana", "cherry", "date"])

|

["apple", "banana", "cherry", "date"]

#include <bits/stdc++.h>

using namespace std;

vector<string> merge\_sort(vector<string> a, int f, int l)

{

if (f == l)

{

vector<string> last\_1= {a[f]};

return last\_1;

}

int m = (f + l) / 2;

vector<string> b = merge\_sort(a, f, m);

vector<string> c = merge\_sort(a, m + 1, l);

vector<string> last;

int i = 0, j = 0;

cout<<"19 ->"<<"l :"<< last.size()<<endl;

cout<<"20"<<":"<< b.size()<<endl;

cout<<"21"<<":"<< c.size()<<endl;

while (i < b.size() && j < c.size())

{

if (b[i] <= c[j])

{

last.push\_back(b[i]);

i++;

cout<<"30->"<<"l :"<< last.size()<<endl;

}

else

{

last.push\_back(c[j]);

j++;

cout<<"37->"<<"l :"<< last.size()<<endl;

}

}

cout<<"40-> "<<"l :"<< last.size()<<endl;

cout<<"41"<<":"<< b.size()<<endl;

cout<<"42"<<":"<< c.size()<<endl;

while (i < b.size())

{

last.push\_back(b[i]);

i++;

}

cout<<"49-> "<<"l :"<< last.size()<<endl;

cout<< "50"<<":"<< b.size()<<endl;

cout<< "51"<<":"<< c.size()<<endl;

while (j < c.size())

{

last.push\_back(c[j]);

j++;

}

cout<<"58-> "<<"l :"<< last.size()<<endl;

cout<< "59"<<":"<< b.size()<<endl;

cout<< "60"<<":"<< c.size()<<endl;

return last;

}

// Main function to read input and call merge sort

int main()

{

int n;

cin >> n;

vector<string> a(n);

for (int i = 0; i < n; i++)

{

cin >> a[i];

}

int first\_idx=0;

int last\_idx=n-1;

vector<string> sorted\_a = merge\_sort(a, first\_idx, last\_idx);

for (int i = 0; i < n; i++)

{

cout << sorted\_a[i] << " ";

}

return 0;

}