External Sort

Assume that all the data to be sorted are available at one time in internal memory, such as in an array. To sort data stored in an external file, you must first bring the data to the memory and then sort it internally. However, if the file is too large, all the data in the file cannot be brought to memory at one time. You can sort data in a large external file using an external sort.

Assume that a large data with int values is stored in a binary file named verybigdata.dat. This file was created using the program below:

```
#include <iostream>
#include <fstream>
#include <cstdlib>
using namespace std;

int main() {
	srand((unsigned)time(NULL));
	const int dataSize = 800004;
	int data;
	ofstream output("verybigdata.dat", ios::out | ios::binary);
	if (!output) {
		cout << "Cannot open file!" << endl;
		return 1;
	}
	for (int i = 0; i < dataSize; i++)
	{
		data =(rand() % 1000000);
		output.write((char*)&data, sizeof(data));
	}
	output.close();
	if (!output.good()) {
		cout << "Error occurred at writing time!" << endl;
		return 1;
	}

	ifstream input("verybigdata.dat", ios::in | ios::binary);

	int readData;
	cout << "Display first 100 values:" << endl;
	for (int i = 0; i < 100; i++)
	{
		input.read((char*)&readData, sizeof(readData));
		cout << readData << " ";
	}
	input.close();
}
```

A variation of merge sort can be used to sort this file in two phases:

Phase 1: Repeatedly bring data from the file to an array, sort the array using an internal sorting algorithm, and output the data from the array to a temporary file. This process is shown in Figure 1. Ideally, you want to create a large array, but its maximum size depends on how much memory is allocated. Assume that the maximum array size is 100000 int values. In the temporary file, every 100000 int values are sorted. They are denoted as S_1, S_2, … and S_n, where the last segment, S_n may contain less than 100000 values.

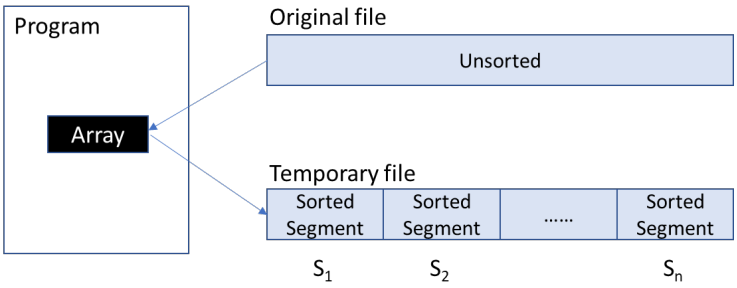

Figure 1: The original file is sorted in segments.

Phase 2: Merge a pair of sorted segments (e.g., S_1 with S_2, S_3 with S_4,…, and so on) into a larger sorted segment and save the new segment into a new temporary file. Continue the same process until only one sorted segment results. Figure 2 shows how to merge eight segments.

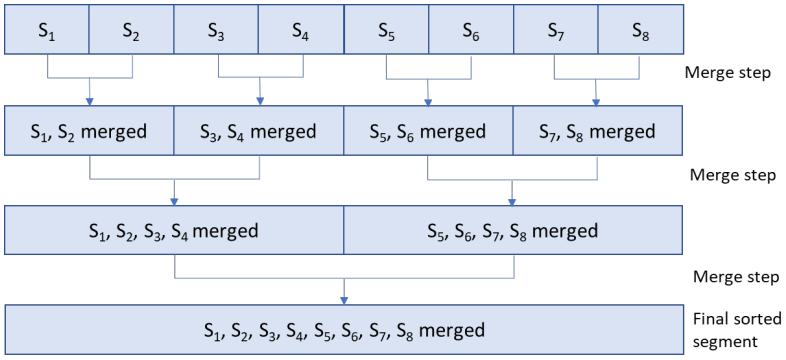

Figure 2: Sorted segments are merged iteratively.

Complete your program for sorting int values in verybigdata.dat and storing the sorted data in sortedfile.dat.