

Basic idea:

1. Generate an array in random, which the size should be a random variable we can define.
2. You should use Brute Force and Divide-And-Conquer respectively to solve maximum subarray problem. Compare the running time with the same input array but the different size. The size of array should include 100, 1000, 10000 and 100000, and you should attach the screenshot of the running time in your report.
3. Compare the result of your program, and see if the time complexity is $O(f(n))$ as mentioned in class, where $f(n)$ is the corresponding complexity to Brute Force and Divide-And-Conquer. Please discuss in your report.
4. Upload format There are three file you need to upload:
student ID_name_HW1.cpp or .c,
student ID_name_HW1.exe,
student ID_name_HW1.doc or .pdf

Program description:

1. Let input be the size of array. As mentioned in 2., you should allow TAs to input different size.
2. After inputting the size of array, randomly generate the elements in the array. For simplicity, you may set the array elements as integers.
3. Process the array with Brute Force and Divide-And-Conquer mentioned on class, and record the processing time needed for maximum subarray search.
4. You shall print some simple instructions or leave some comment, so TAs could realize how to demo your program.

Note that:

You may refer to the textbook or pseudo code to make your project.

By the way, don't worry about getting grade zero if you present incorrect results. You could discuss more comprehensively from the results, and give us some conclusion even the results are incorrect.

Ps.You can “ secretly ” google “Introduction to algorithms, Third Edition pdf” and click third link.