Lab 2 – due 9/14 at 11pm

Analysis of algorithms

1. a. (25 pts) Compile and run the ThreeSumDoubling application in the Example14-5 folder. Show the screenshot of it running and the times printed.
   b. (10 pts) Create a text file that contains:
      i. the times that you received as output and compare it to my times in the 1.4 analysis cont (week3) lecture slides (slide 38). Compare both for all the N's and the time per instruction (last line in output).
      ii. an estimate of how long it would take on your computer to run ThreeSum with N=16000 (using that time that you printed at the end of time of one instruction).
2. (5pts each) In that same text file, give the time equation and the time order of growth (Big O) as a function of n) for each following code fragments (don't use the f to g example – run as code if you need to):

   i.

   ```
   int sum = 0;
   for (int k = n; k > 0; k /= 2)
       for (int i = 0; i < k; i++)
           sum++;
   ```

   ii.

   ```
   int sum = 0;
   for (int i = 1; i < n; i *=2))
       for (int j = 0; j < i; j++)
           sum++;
   ```

   iii.

   ```
   int sum = 0;
   for (int i = 1; i < n; i *=2)
       for (int j = 0; j < n; j++)
           sum++;
   ```

3. (50 pts) Create a program that, given an array a[] of n double values, finds a closest pair: two values whose difference is no greater than the difference of any other pair (in absolute value). The running time of your program should be linearithmic in the worst case. Must write your own. No internet code, but you can start with predefined Java data structures. Your output should give this information:

   ```
   The numbers 708.074085 and 708.074104 are the closest pair with a difference of 0.000019
   ```

   Use the file lab2in.txt (in d2l folder with the lab write up) as your input. Print the difference to 6 numbers after the decimal.

   Submission:

   You will submit a text file for 1b and 2a, b, and c. You will submit a source file (program 3 only) and screenshot for each program (problems 1a and 3).

Rubric for program (1a):

- 25 points – submitted screen shot of run with output, showing your program runs correctly
- 25 points – YOU MUST USE THE CODE IN EXAMPLE 14-5 in d2l so make sure to submit the source code as you have modified it to work in your IDE

Rubric for program (3):

- 5 points – style – used comments and proper indention
- 5 points – comment in code showing the Big O of your program and how you came to that conclusion
- 15 points – your program is Big O linearithmic time as worst case
- 5 points – submitted screenshot of the run
- 5 points – program runs correctly
- 5 points – output showed the difference to 6 numbers after the decimal point
- 10 points – the file called lab2in.txt (in d2l) was used as your input; you read it in – did not hardwire it into your code
- You will get a 0 if you:
  a) Do not submit the source code
  b) Do not write the program from scratch (but you can use java predefined data structures)