Cg633 Chenxin Guo
You could find my repo from this link:
https://github.coecis.cornell.edu/cg633/ORIE5270.git

Q1:
The first is to setup new heap every test you run
Then the first test: insert 3 people's name and then test the length of the nameheap: whether they equal to 3
Second test: first insert 3 people's names, then return the smallest name (the smallest rank of the alphabetical order)
Third test: return False if the there is no name in the nameheap when call function smallestName
Forth test: return True if contain the name in the nameheap
Fifth test: first insert 3 people's names and then delete the name according to alphabetical order, therefore, this test should return 'Ben Grimmer'
Last test: return False if the there is no name in the nameheap when call function deleteSmallestName

Q2

(a) The subnormal of a double can be written as 0.B*2^(A-1022) and A=0, since the smallest B is 2^(-52), so any absolute value of the number between 2^(-52-1022) and 2^(-1022) is also a subnormal number. so 2^(-51-1022) is also a subnormal

(b) The function is to judge whether the abs(number) is between 2^(-52-1022) and 2^(-1022), if yes return True, else return False. And I in the test part, I used several numbers to test that, the first two should be the subnormal number and the latter 3 are not. So the correct test results is: true, true, false, false, false.

Q3

(1) I make a within function that return whether abs((a - b)/b) <= epsm, so if correctly, then the test result should be all true.

(2) when b= range(10, 200, 10) and c=1, and when b<100, the sum and product are all return True, but after 100, from then on, the result of the test, x1+x2 always pass the test, but x1*x2 always fail the test. The reason is that: from the root formula: -b-sqrt(b^2-4ac)/2a and : -b+sqrt(b^2-4ac)/2a, if c is small relative to b, then we could ignore 4ac, and x1=b/a and x1=0, but when they are added together, the effect will offset and therefore, the error is still small. If they are multiplied together, the difference will be huge.

(3) The results show that if I change both of the x into this new formula, then the test result is the same in my origin test. When b is bigger than 100, then the result start to report False.

Q4

In order to show the time is O(n^2), I choose the shape of A from (20000, 20000) to (40000, 40000), so that from the plot, we could see the difference of time.

I first record the time before the calculation, then after the matric calculation, I record again the time. So the time delta is the time 2 – time 1.

However, from the direct plot, it is very hard to see whether the time plot is quadratic, so I do a transformation: math.sqrt the time and later plot the time. If we see from the plot that it looks lt looks like a straight line go start from origin, then we can say that the calculation of this is O(N^2) time.