2016 fall Algorithm Midterm

(10%)1. Proof that f(n)=0(g(n)) iff f(n)=Ω(g(n)) and f(n)=O(g(n)).

(10%)2.f(n)=10000n^2+200n+1, calculate the O(f(n)), 0(f(n)) and Ω(g(n)).

(10%)3.prove log(n!)= 0(nlogn).

(10%)4.prove or disprove the following statement: find the largest number in a list of n numbers requires at least n-1 comparisons. Give reason for your answer.

(10%)5.Illustrate the QUICK-SORT and analysis its time complexity.

(10%)6.List all the comparison sort that the average case is O(nlogn).

(10%)7.Show that we can find the both of minimum and maximum using at most $3\lfloor n/2 \rfloor$ comparisons.

(10%)8.Show that when all elements are distinct, the best-case running time of HEAPSORT is Ω(nlogn).

(10%)9.Prove that COUNTING-SORT is stable.

(10%)10.Illustrate the operation of HEAP-SORT and show the the BUILD-HEAP complexity can be O(n).