**Assignment 1**

1. a) The loop body of i runs in O(1), and executed O(n) times, the loop body of j runs in O(n), and executed O(1) times, the loop body of k runs in O(1), and executed O (1) times, so the total running time is O(1)\*O(1)\*O(n)=O(n).

b) The loop body of i runs in O(1), and executed O(n-1) times,

the loop body of j executed O (log2n), the loop body of k executed O(log2n), so the total running time is O(n-1)\* O (2log2n)=O(n\*log2n)

c) The loop body of j runs in O(1), and executed O(n) times in the inner while loop, the loop body of i runs O(n) times in the outer while loop, so the total running time is O(n)\*O(n)=O(n2)

d) The inner loop of i runs (n/2) times, and the outer loop of i runs (n/2) times, so the total running time is O(n).

1. a)

The total times of compare is: n-1, so T(n)=O(n).

b)

The worst case is there exist one couple and Bob finds them at the final comparison, the total times of comparison is: (n-1) \* (n-2) \* … \* 2 \* 1=n(n-1)/2, so T(n)=O( n2 )

1. a) 257

b) 51

c) 336

d) 17

1. a) O ( n3 )

b) O (n2)

c) O (n\*logn)

d) O ( n**2**)

1. input array[]

int len is the length of array[]

int new\_arr[ len ]，j=0

for int i in range len

if array[i] != every element in new\_arr

new\_arr[j]=array[i]

j++

int final\_arr[j-1]

copy value in new\_arr[0, j-1] to final\_arr

return final\_arr



int bestrewards(int array[],int len){

int i=0, j, test, max=array[0];

while (i<len){

test =0;

for (j=i, j<len, j++){

test += array[j];

if (test>max){

max = test;

}

}

}

return max;

}