4(a).

```
def closest(l:[int])->int:
 a = set()                                  O(N)
 for i in range(len(l)):                    O(N)
   for j in range(len(l)):                  O(N)
     if i != j:                             O(1)
       a.add(abs(l[i]-l[j]))                O(1)
 return min(a)                              O(N*N)
```

(b)

O(N) +O(N*N*2) + O(N*2)

(c) O(N^2)

```
def closest(l:[int])->int:
  a = sorted(l)                             O(N Log N)
  m = -1                                    O(1)
  for i in range(len(a)-1):                 O(N)
    if m==-1 or abs(a[i+1]-a[i])<min:       O(1)
       m = abs(a[i+1]-a[i])                 O(1)
  return m;                                 O(1)
```

(b)

O(N Log N) + O(1) + O(N*2) + O(1)

(c) O(N Log N)

5. (a)

(1) Tn(N) = c * N * (Log N)^3

When N=1000000, Tn(N) is 80s.

So

c = 80 / (1000000 * (Log 1000000)^3) = 1.01033536e-08

(b)

(2)

Tn(1000000000) = 1.010335e-08 * 80 / (1000000000 * (Log 1000000000)^3)) = 270000

6.

N = Problem Size	Complexity Class	Time to Solve on Old Machine (secs)	M Solvable in the same Time on a New Machine 2x as Fast
10^6	$O(Log_2 N)$	1	$c*Log_2 10^6=1 \Rightarrow c = 1/Log_2 10^6$ $(1/2)*c*Log_2 N = 1.\ N = 4 * 10^6$
10^6	$O(N)$	1	$N = 2 * 10^6$
10^6	$O(N\ Log_2 N)$	1	$c*10^6*Log_2 10^6=1 \Rightarrow c=1/(10^6*Log_2 10^6)$ $(1/2)*c*N*Log_2 N=1.\ N=3656807$
10^6	$O(N^2)$	1	$c*N^2=1 \Rightarrow c = 1/10^{12}$ $(1/2)*c*N2=1.$ $N=2^{1/2}*10^6$