Q3) (a) 1

```
Public static double addOverNs (int n) {
if (n == 1) return 1;
double number = 1.0/n;
  if (n%2 == 0) sum += number //sum is a global variable
  else sum -= number
  retur sum + addOverNs (n-1);
```

(2)

(a)

```
Public Void recursiveCubes (int n) {
if (n == 1) {
  system.out.println ((int) n);
    return; }
  recursiveCubes (n-1);
  system.out.Println((int) Math.Pow(n, 3));
  }
```

(3) (a) $2T(n-1)+n$

$= 2[2T(n-2)+n-1]+n$

$= 2^2 T(n-2) + 2^1(n-1) + 2^0 n$

$= 2^2[2T(n-3)+(n-2)] + 2^1(n-1) + 2^0 n$

$= 2^3 T(n-3) + 2^2(n-2) + 2^1(n-1) + 2^0(n-0)$

$= 2^k + 2^{k-1}(n-(k-1)) + 2^{k-2}(n-k+2) + \cdots 2^{n-k}(n-k)$

$= 2^k + 2^{k-1}(1) + 2^{k-2}\cdot 2 + 2^{k-3}\cdot 3 \cdots 2^0(0)$

$= 2^k + 2^{k-1} + 2\,2^{k-2} + 3\,2^{k-3} \cdots 0\,2^{k-n}$

$= 2^{nk} + \sum_{i=0}^{n} i\,2^i \Rightarrow 2^k + \sum_{i=0}^{n} i\,2^i$

$= 2^{kn} + \frac{n\,2^{n+2} - n\,2^{n+1} - 2^{n+1} + 2}{1} = \Theta(n\,2^{n+2})$

(b) $4[4T(\frac{n}{4}) + \frac{n}{2}] + n$

$= 16T(\frac{n}{4}) + 2n + n$

$= 16[4T(\frac{n}{8}) + \frac{n}{4}] + n + 2n$

$= 64T(\frac{n}{8}) + 4n + 2n + n$

$= 2^3 T(\frac{n}{8}) + 4n + 2n + n$

when $\frac{n}{8^k} = 1 \Rightarrow 2^k T(\frac{n}{k}) + \frac{kn}{2} + \frac{kn}{4} \cdots \frac{kn}{k}$

$= 2^k + \frac{kn}{2} + \frac{kn}{8^4} + \frac{kn}{8^8} \cdots + n$

but $\frac{n}{n} = 1$ ie, $n = k$

$= 2^n + \frac{n^2}{2^1} + \frac{n^2}{2^3} + \frac{n^2}{2^8} \cdots \frac{n^2}{2^n}$

$= 2^n + n^2 \sum_{i=1}^{n} \frac{1}{2^i} = 2^n + n^2 \sum_{i=0}^{n} 2^{-i}$

$O(2^n)$