Solution to HW 3

February 27, 2013

1. $T(n) = T(\sqrt{n}) + 2n$
   Let n be 2^k
   So $T(2^k) = T(2^{k/2}) + 2.2^k$
   Let $(T(2^k) = S(K))$
   $S(K) = S(K/2) + 2^{K+1}$
   Applying Master's Theorem:
   a=1, b=2
   $2^{n+1/2} = K.2^{n+1}$
   $K = 1/2$
   So T(n)=T$(2^k) = \theta(2^k)$
   T(n)=$\theta(n)$

2. $T(n) = T(n-c) + T(c) + f(n)$

   (a) $T(n) = log(n) + [log(n-c) + log(c)] + [log(n-2c) + log(c)] + ... + [log(n-lc) + log(c)]$
   $= T(n) = log(n) + log(n-c) + log(n-2c) + ... + log(n-lc) + (k-1).log(c)$
   $= O(nlogn)$

   (b) $T(n) = n + (n-c) + ... + (n-kc) + .. + 1$
   $= n^2 - (1 + 2 + 3 + .. + n)c$
   $= O(n^2)$

3. $(a+bi)(c+di) = (ac-bd) + (ad+bc)i$
   As we can see above there are 3 multiplications required at each step.
   So at each node of the tree we are using 3 multiplications This translates to :
   $T(n) = 3 + 2 * 3 + 2 * 3^2 + ..2 * 3^n$
   $T(n) = 2.T(n/2) + 3$
   Solving this recurrence we get:
   $T(n) = 3\sum_{i=0}^{l} 2^i$
   $= 3(2^{k+1} - 1)$

$3(n + 1 - 1)$
$= 3n$
$T(n) = O(n)$