Algorithm 2020 Spring: Assignment Week 2

Due on Monday, March 3, 2020

李胜锐 2017012066

Question 1

Solve the Recursions

(a) $T(n) = 2T(\sqrt{n}) + 1$

Let $m = \lg n$, then $T(2^m) = 2T(2^{m/2}) + 1$

Let $T(2^m) = S(m)$, then $S(m) = 2S(m/2) + 1$

Thus, $m^{\log_a b} = m^{\log_2 2} = m$, $f(m) = 1 = O(m^{1-\epsilon})$, when $\epsilon = 0.1$

Then $S(m) = \Theta(m), T(2^n) = \Theta(n), T(n) = \Theta(\lg n)$

(b) $nT(n) = (n-2)T(n-1) + 2$

Assume that $T(n) = O(1)$

Suppose $T(n-1) < c$, then $nT(n) = (n-2)T(n-1) + 2 < c(n-2) + 2 = cn - 2c + 2$

$T(n) < c - \frac{2c-2}{n}$

Thus, let $c = 1$. We can get $T(n) < c$.

So it is true that $T(n) = O(1)$

Noticed that $T(n) = \Omega(1)$

Therefore, $T(n) = \Theta(1)$