CS301$_h w1$

Mustafa Yağız Kılıçarslan

October 2022

1 Recurrences (10 points)

Give an asymptotic tight bound for T(n) in each of the following recurrences. Assume that T (n) is constant for n 2. No explanation is needed.

Note that according to Master Method:
if recurrence is in the form of $T(n) = aT(n/b) + f(n)$ where $a, b \geq 1$ and $f(n)$ is asymptotically positive, then:

Case 1: *if* $f(n) = O(n^{\lg_b a-\epsilon})$ for some $\epsilon > 0$ *then* $T(n) = \Theta(n^{\lg_b a})$

Case 2: *if* $f(n) = \Theta(n^{\lg_b a})$ *then* $T(n) = \Theta(n^{\lg_b a} \lg n)$

Case 3: *if* $f(n) = \Omega(n^{\lg_b a+\epsilon})$ for some $\epsilon > 0$ and if $cf(n) \geq af(n/b)$ for some $c < 1$ and for large n *then* $T(n) = \Theta(f(n))$

a. $T(n) = 2T(n/2) + n^3$
Using Master Theorem **Case 3**:
$a = b = 2 \geq 1$, $f(n) = n^3$ asymptotically positive
$f(n) = n^3 = \Omega(n^{\lg_2 2+\epsilon}) = \Omega(n^\epsilon)$, $\epsilon > 3$
Check: $2(n/2)^3 = n^3/4 \leq cn^3$ for $c = 1/4 < 1$ and for large n
In this case we can apply Case 3:
$T(n) = \Theta(f(n)) = \Theta(n^3)$

b. $T(n) = 7T(n/2) + n^2$ Using Master Theorem **Case 1**:
$a = 7 \geq 1$, $b = 2 \geq 1$, $f(n) = n^2$ asymptotically positive
$f(n) = n^2 = O(n^{\lg_2 7-\epsilon})$, $\epsilon > 0$
We can apply Case 1:
$T(n) = \Theta(n^{\lg_b a}) = \Theta(n^{\lg_2 7})$

c. $T(n) = 2T(n/4) + \sqrt{n}$
Using Master Theorem **Case 2**:
$a = 2 \geq 1$, $b = 4 \geq 1$, $f(n) = \sqrt{n}$ asymptotically positive

$n^{\lg_b a} = n^{\lg_4 2} = \sqrt{n}$
$f(n) = \sqrt{n} = \Theta(n^{\lg_4 2}) = \Theta(\sqrt{n})$
We can apply Case 2:
$T(n) = \Theta(n^{\lg_b a} \lg n) = \Theta(\sqrt{n} \lg n)$

d. $T(n) = T(n-1) + n$
Prove that $T(n) = \Theta(n^2)$

d.1 Prove $T(n) = O(n^2)$ by induction:
Inductive Step: Assume that $T(k) \leq ck^2$ for some $c > 0$, for all $k < n$ holds.

$$T(n) \leq c(n-1)^2 + n$$
$$T(n) \leq cn^2 - 2cn + c + n$$
$$T(n) \leq cn^2 - (2cn - c - n)$$

$(2cn - c - n) > 0$ for $c > 1/2$ and $n_0 < \frac{c}{2c-1}$
Therefore, $T(n) \leq cn^2$
$T(n) = O(n^2)$ for mentioned c and n_0 values.

d.2 Prove $T(n) = \Omega(n^2)$ by induction:
Inductive Step: Assume that $T(k) \geq ck^2$ for some $c > 0$, for all $k < n$ holds.

$$T(n) \geq c(n-1)^2 + n$$
$$T(n) \geq cn^2 - 2cn + c + n$$
$$T(n) \geq cn^2 + (n + c - 2cn)$$

$(2cn - c - n) > 0$ for $c < 1/2$ and $n_0 > \frac{c}{2c-1}$
Therefore, $T(n) \geq cn^2$
$T(n) = \Omega(n^2)$ for mentioned c and n_0 values.

Since we can show that: $T(n) = O(n^2)$ and $T(n) = \Omega(n^2)$
$T(n) = \Theta(n^2)$

2 (Longest Common Subsequence - Python)

i. What is the best asymptotic worst-case running time of the naive recursive algorithm shown in Figure 1? Please explain.

ii. What is the best asymptotic worst-case running time of the recursive algorithm with memoization, shown in Figure 2? Please explain