CSORW4231 HOMEWORK 1

Due Mon, Feb 06

Jun Hu
(jh3846)

Problem 1 (10 points)**.** Exercise 3.1-1 of the textbook (Page 52).

Solution.

Since $f(n)$ and $g(n)$ are asymptotically nonnegative:

$$\begin{cases} 0 \le f(n) \le max(f(n), g(n)) \le f(n) + g(n) & (1) \\ 0 \le g(n) \le max(f(n), g(n)) \le f(n) + g(n) & (2) \end{cases}$$

By adding the two inequalities (1) and (2):

$$0 \le f(n) + g(n) \le 2max(f(n), g(n)) \le 2(f(n) + g(n))$$

Therefore: $\exists\ c_1 = \frac{1}{2},\ c_2 = 1,\ n_0 > 0,\ \forall\ n \ge n_0$, s.t.

$$0 \le c_1(f(n) + g(n)) \le max(f(n), g(n)) \le c_2(f(n) + g(n))$$

By definition:

$$max(f(n), g(n)) = \Theta(f(n) + g(n))$$

□

CSORW4231 HOMEWORK 1

Due Mon, Feb 06

Jun Hu
(jh3846)

Problem 2 (10 points)**.** Exercise 3.1-5 (Page 53).

Solution.

From $f(n) = \Omega(g(n))$, we get $\exists\ c_1 > 0,\ n_1 > 0,\ \forall\ n \geq n_1$, s.t.

$$0 \leq c_1 g(n) \leq f(n)$$

From $f(n) = O(g(n))$, we get $\exists\ c_2 > 0,\ n_2 > 0,\ \forall\ n \geq n_2$, s.t.

$$0 \leq f(n) \leq c_2 g(n)$$

Let $n_0 \geq n_1 > 0$ and $n_0 \geq n_2 > 0$, we get $\exists\ c_1 > 0,\ c_2 > 0,\ n_0 > 0,\ \forall\ n \geq n_0$, s.t.

$$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$$

By definition:

$$f(n) = \Theta(g(n))$$

Also, from $f(n) = \Theta(g(n))$, we get $\exists\ c_3 > 0,\ c_4 > 0,\ n_3 > 0\ \forall\ n \geq n_3$, s.t.

$$0 \leq c_3 g(n) \leq f(n) \leq c_4 g(n)$$

Separately:

$$0 \leq c_3 g(n) \leq f(n) \qquad \Rightarrow \qquad f(n) = \Omega(g(n))$$
$$0 \leq f(n) \leq c_4 g(n) \qquad \Rightarrow \qquad f(n) = O(g(n))$$

□

CSORW4231 HOMEWORK 1

Due Mon, Feb 06

Jun Hu
(jh3846)

Problem 3 (10 points)**.** Problem 3-1(a), (b) and (c) (page 61).

Solution.

(a) From $k \geq d$, $a_d > 0$:

$$\lim_{n\to\infty} \frac{p(n)}{n^k} = \lim_{n\to\infty} \frac{\sum_{i=0}^{d} a_i n^i}{n^k} = \lim_{n\to\infty} \frac{a_0 + a_1 n + \cdots + a_{d-1}n^{d-1} + a_d n^d}{n^k}$$
$$= \lim_{n\to\infty} \left(\frac{a_0}{n^k} + \frac{a_1}{n^{k-1}} + \cdots + \frac{a_{d-1}}{n^{k-d+1}} + \frac{a_d}{n^{k-d}} \right) = L \ (0 \leq L \leq a_d)$$

Which is, $\exists\ \delta > 0$, $\forall\ n > n_0$, s.t.

$$\frac{p(n)}{n^k} - L \leq \delta \qquad \Rightarrow \qquad p(n) \leq (L+\delta)n^k \leq (a_d + \delta)n^k$$

Set $\delta = \frac{1}{2}a_d$, $\exists\ c = \frac{3}{2}a_d > 0$, $n_0 > 0$, $\forall\ n > n_0$, s.t.

$$0 \leq p(n) \leq cn^k \qquad \Rightarrow \qquad p(n) = O(n^k)$$

(b) From $k \leq d$, $a_d > 0$:

$$\lim_{n\to\infty} \frac{p(n)}{n^k} = \lim_{n\to\infty} \frac{\sum_{i=0}^{d} a_i n^i}{n^k} = \lim_{n\to\infty} \frac{a_0 + a_1 n + \cdots + a_{d-1}n^{d-1} + a_d n^d}{n^k}$$
$$= \lim_{n\to\infty} \left(\frac{a_0}{n^k} + \frac{a_1}{n^{k-1}} + \cdots + \frac{a_{d-1}}{n^{k-d+1}} + \frac{a_d}{n^{k-d}} \right) = L \ (a_d \leq L \to \infty)$$

Which is, $\exists\ \delta > 0$, $\forall\ n > n_0$, s.t.

$$\frac{p(n)}{n^k} - L \geq -\delta \qquad \Rightarrow \qquad p(n) \geq (L-\delta)n^k \geq (a_d - \delta)n^k$$

Set $\delta = \frac{1}{2}a_d$, $\exists\ c = \frac{1}{2}a_d > 0$, $n_0 > 0$, $\forall\ n > n_0$, s.t.

$$p(n) \geq cn^k \geq 0 \qquad \Rightarrow \qquad p(n) = \Omega(n^k)$$

(c) From $k = d, a_d > 0$:

$$\lim_{n\to\infty} \frac{p(n)}{n^k} = \lim_{n\to\infty} \frac{\sum_{i=0}^{d} a_i n^i}{n^k} = \lim_{n\to\infty} \frac{a_0 + a_1 n + \cdots + a_{d-1}n^{d-1} + a_d n^d}{n^k}$$
$$= \lim_{n\to\infty} \left(\frac{a_0}{n^k} + \frac{a_1}{n^{k-1}} + \cdots + \frac{a_{d-1}}{n^{k-d+1}} + \frac{a_d}{n^{k-d}} \right) = a_d$$

Which is, $\exists\ \delta > 0$, $\forall\ n > n_0$, s.t.

$$\left| \frac{p(n)}{n^k} - a_d \right| \leq \delta \qquad \Rightarrow \qquad (a_d - \delta)n^k \leq p(n) \leq (a_d + \delta)n^k$$

Set $\delta = \frac{1}{2}a_d$, $\exists\ c_1 = \frac{1}{2}a_d > 0$, $c_2 = \frac{3}{2}a_d > 0$, $n_0 > 0$, $\forall\ n > n_0$, s.t.

$$0 \leq c_1 n^k \leq p(n) \leq c_2 n^k \qquad \Rightarrow \qquad p(n) = \Theta(n^k)$$

$\square$

CSORW4231 HOMEWORK 1

Due Mon, Feb 06

Jun Hu

(jh3846)

Problem 4 (10 points)**.** Problem 3-4(b), (e) and (f) (Page 62).

Solution.

(b) False.
Counterexample: Set $f(n) = n, g(n) = n^2$, So $min(f(n), g(n)) = min(n, n^2) = n = f(n)$, but

$$f(n) + g(n) = n + n^2 = \Theta(n^2) = \Theta(g(n)) \neq \Theta(min(f(n) + g(n))$$

(e) False.
Counterexample: When $0 < f(n) < 1$, $(f(n))^2 < f(n)$, e.g. $f(n) = \frac{1}{n}$ s.t.

$$f(n) = \omega((f(n))^2)$$

(f) True.
From $f(n) = O(g(n))$: $\exists\ c > 0$, $n_0 > 0$, $\forall\ n > n_0$, s.t.

$$f(n) \leq cg(n)$$

Set $c' = \frac{1}{c}$, $\exists\ c' > 0$, $n_0 > 0$, $\forall\ n > n_0$, s.t.

$$g(n) \geq c'f(n) \qquad \Rightarrow \qquad g(n) = \Omega(f(n))$$

□

CSORW4231 HOMEWORK 1

Due Mon, Feb 06

Jun Hu
(jh3846)

Problem 5 (10 points)**.** Exercise 2.3-7 (Page 39).

Solution.

```
1  EXISTENCE(S, x) {
2      A = MERGE-SORT(S)          // Let A be set S in nondecreasing order
3      i = 1
4      j = n
5      while i < j                // Test sum from ends of the sorted set
6        if A[i] + A[j] == x
7          return true            // Find the two elements
8        else if A[i] + A[j] < x
9          i = i + 1
10       else
11         j = j - 1
12     return false               // Not find the two elements
13 }
```

For line 2, MERGE-SORT costs $\Theta(nlgn)$;
For line 3 - 4, initial assignment costs constant time $\Theta(1)$;
For line 5, while loop condition costs $O(n)$;
For line 6 - 11, while loop comparison body costs $O(n)$;
For line 12, final return cost constant time $\Theta(1)$.
Apparently, $\Theta(nlgn)$ is dominant in the algorithm. In sum, the runtime is $\Theta(nlgn)$.

□

CSORW4231 HOMEWORK 1

Due Mon, Feb 06

Jun Hu
(jh3846)

Problem 6 (10 points)**.** Problem 2-3 (Page 41). Skip (b), but do take a minute to think about the naive implementation. Also if you are not familiar with induction, work on (c) and (d) after next Mondays class.

Solution.

(a)

$$\begin{aligned} T(n) &= T(n-1) + c \qquad (c > 0 \text{ is a constant }) \\ &= T(n-2) + 2c = T(n-3) + 3c = \cdots \\ &= T(0) + cn \qquad (T(0) > 0 \text{ is a constant }) \\ &= \Theta(n) \end{aligned}$$

(c) Basis: $i = n$, $y = \sum_{k=0}^{-1} a_{k+n+1}x^k = 0$ True.
Induction steps:
At the start of the i-th iteration in line 2 according to the loop invariant, s.t.

$$y = \sum_{k=0}^{n-(i+1)} a_{k+i+1}x^k$$

After perform line 3, s.t.

$$\begin{aligned} y = a_i + x\left(\sum_{k=0}^{n-(i+1)} a_{k+i+1}x^k\right) &= a_i + \sum_{k=0}^{n-(i-1)} a_{k+i+1}x^{k+1} \\ &= \sum_{k=0}^{n-((i-1)+1)} a_{k+(i-1)+1}x^k \end{aligned}$$

Which is the $(i-1)$-th iteration in line 2 according to the loop invariant.
Conclude: $i = 0$, perform line 3, s.t.

$$y = a_0 + x\left(\sum_{k=0}^{n-1} a_{k+1}x^k\right) = \sum_{k=0}^{n} a_k x^k$$

Then, $i = -1$, the loop terminates.

(d) From the analysis in (c), the code fragment exactly obtains the result of the polynomial $P(x)$ characterized by the coefficients $a_0, a_1, \ldots, a_n$.

□