Solutions for Assignment 1

CSCI2100B

October 16, 2013

1 Written Assignment

Exercise 1.1

(6) $\sum_{i=1}^{n} ia^i$
Solution: According to the summation formula of geometric progression,
$T = \sum_{i=1}^{n} a^i = \frac{a^{n+1}-a}{a-1}$.
Let $S = \sum_{i=1}^{n} ia^i$ $aS = \sum_{i=1}^{n} ia^(i+1) = \sum_{i=1}^{n}(i-1)a^i + na^{n+1}$.
$(a-1)S = na^{n+1} - T \Rightarrow S = \frac{1}{a-1}(na^{n+1} - \frac{a^{n+1}-a}{a-1})$.
(8) $\sum_{i=0}^{n} i^2$
Solution:

$$\begin{array}{rc} & i^3 - (i-1)^3 = 3i^2 - 3i + 1 \\ \Rightarrow & n^3 = \sum_{i=1}^{n}(i^3 - (i-1)^3) = \sum_{i=1}^{n}(3i^2 - 3i + 1) \\ = & 3\sum_{i=1}^{n} i^2 - 3\frac{n(n+1)}{2} + n \\ \Rightarrow & \sum_{i=0}^{n} i^2 = \sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} \end{array}$$

(12) Is $2^{n+1} = O(2^n)$?
Solution: Yes. $2^{n+1} = 2 \cdot 2^n = O(2^n)$
(13) Is $2^{2n} = O(2^n)$?
Solution: No. $2^{2n} = O(4^n)$

Exercise 1.2

(3) If $GCD(a,b) = p$ and $GCD(c,d) = q$, is $GCD(ac,bd) = pq$ true for all the a,b,c,d? Either prove it or give a counterexample.
Solution: Counterexample: a = 5, b = 4, c =4, d = 5.

Exercise 1.3

(1) $T(n) = aT(n-1) + bn, T(1) = 1$
Solution: First consider the problem of solving $T(n) = aT(n-1) + b$. An intuitive approach is to let $S(n) = T(n) + k$ and then replace $T(n)$ with $S(n)$ in the original formula to make it $S(n) = aS(n-1)$.
Here the problem changes to $T(n) = aT(n-1) + bn$. We now let $S(n) = T(n) + pn + q$. After replacing $T(n)$ with $S(n)$, we obtain $S(n) - pn - q = a(S(n-1) - p(n-1) - q) + bn$, which leads to $S(n) = aS(n-1) + (p - ap + b)n + (q + ap - aq)$. Thus we let $p - ap + b = q - ap - aq = 0$, we can get $S(n) = aS(n-1)$. After calculation, $p = \frac{b}{a-1}$ and $q = \frac{ab}{(a-1)^2}$. Since we know $S(1)$ according to $T(1)$ and we have $S(n) = aS(n-1)$, we can solve $S(n)$ easily. Finally, we use

$S(n) = T(n) + pn + q$ to get $T(n)$.
One thing you should take care is that when $a = 1$, $T(n) = T(n-1) + bn$. Then $T(n) = T(1) + 2b + 3b + ... + nb = 1 + \frac{(n+2)(n-1)}{2}b$.
(5) Solve $x_n = x_{n-1} - \frac{1}{4}x_{n-2}$, with $x_0 = 1, x_1 = 1/2$.
Solution: First solve the quadratic formula $t^2 - t + \frac{1}{4} = 0$.
The solutions are $t_1 = t_2 = \frac{1}{2}$. Thus the solution is of the form $x_n = a(\frac{1}{2})^n$.
To satisfy the initial conditions, we can obtain $a = 1$. Thus, $x_n = (\frac{1}{2})^n$.

Exercise 1.4

(5) Prove $2lg(n!) > nlgn$ by using Induction, where n is a positive integer greater than 2.
Solution: Let $P(n)$ be $lg(n!) > nlgn$, where n is a positive integer.
For $n = 1$, $L.H.S = 2lg(1!) > lg(1) = R.H.S$. P(1) is true.
Assume P(k) is true, i.e. $2lg(k!) > klgk$, where k is a positive integer
For $n = k + 1$,

$$\begin{aligned} \text{L.H.S} &= 2lg((k+1)!) \\ &= 2(lg(k!) + lg(k+1)) \\ &> klgk + 2lg(k+1) && \text{(by assumption)} \\ &> (k-1)lg(k+1) + 2lg(k+1) && (k+1 > e > (1+\tfrac{1}{k})^k \Rightarrow k^k > (k+1)^{k-1} \text{ for } k \geq 2) \\ &= (k+1)lg(k+1) \\ &= \text{R.H.S} \end{aligned}$$

$P(k+1)$ is also true.
Therefore, by M.I., $P(n)$ is true for all positive integer n.
(6) The number generated by the formula $n^2 + n + 17$ is prime for $n \geq 0$, where n is an integer. Either prove it or disprove it by counterexample.
Solution: No. Let $n = 17$. Then $n^2 + n + 17 = 17 \times (17 + 1 + 1) = 17 \times 19$

Exercise 1.6

```
(1) for i = 1 to n;
        for j = 1 to n;
               x := x + 1;
```

Solution: $f(n) = n^2$, $g(n) = n^2$.

```
(3) for i = 1 to n;
        for j = i to n;
            for k = 1 to j;
                x := x + 1;
```

Solution: $f(n) = \sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{k=1}^{j} 1 = \sum_{i=1}^{n}\sum_{j=i}^{n} j = \frac{1}{2}\sum_{i=1}^{n}(n+i)(n-i+1) = \frac{1}{2}\sum_{i=1}^{n}(n(n+1) + i - i^2) = \frac{n(n+1)(2n+1)}{6}$, $g(n) = n^3$.

```
(5) for i = 1 to n;
        for j = i to n;
            for k = 1 to 1000;
                x := x + 1;
```

Solution: $f(n) = \sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{k=1}^{1000} 1 = 1000\sum_{i=1}^{n}\sum_{j=i}^{n} 1 = 1000\sum_{i=1}^{n}(n-i+1) = 1000\sum_{i=1}^{n} i = 500n(n+1)$, $g(n) = n^2$.

Exercise 1.8

(3) What is the running time of this algorithm with the following assumptions? What is its big-O notation?

Statement	**Time Unit**
assignment	1
+	1.25
*	1.75
for-next loop set-up	2.3
each loop	1.5

Solution: First assignment consumes 1 unit. Setting up for-next loop consumes 2.3 units. For each loop, there are one $*$, one $+$, and one assignment. Thus, $(1.75 + 1.25 + 1 + 1.5) * (n + 1) = 5.5n + 5.5$ units will be consumed. In total, the running time is $5.5n + 8.8$ units, which is $O(n)$ in big-O notation.

Exercise 1.9

(2) Calculate the time and space complexity for $n = 10, 20, 30, 50, 70$, and 100 for each algorithm.
Solution: Just pay attention to different ranges.
(4) Come up with a strategy that you would use to minimize the time and space complexity individually?
Solution: $t(n) = min\{t_A(n),\ t_B(n)\}$, $s(n) = min\{s_A(n),\ s_B(n)\}$. Thus,

$$t(n) = \begin{cases} n & \text{if } 1 \le n < 50 \\ n^2 & \text{if } 50 \le n < 70 \\ n^3 & \text{if } 70 \le n \le 100 \end{cases}$$

$$s(n) = \begin{cases} n & \text{if } 1 \le n < 20 \\ 1.5n & \text{if } 20 \le n < 50 \\ 0.5n & \text{if } 50 \le n \le 100 \end{cases}$$

2 Programming Assignment

Exercise 1.17

Analysis: The function isPrime() is responsible for checking the input n is a prime. In the main function, the program uses array P to store all 50 smallest primes. Pay attention to the first for-loop in the main function. The sample code is shown below.

```
#include <stdio.h>
int P[100];
int isPrime(int n) {
    int k;
        if (n <= 1) return 0;
        for (k=2; k*k<=n; ++k)
                if (n%k == 0) return 0;
        return 1;
}
int main() {
        int n, i, k;
        scanf("%d", &n);
        int m = 0;
```

```
        for (i=2; m<50; ++i) {
                if (isPrime(i)) P[m++] = i;
        }
        for (i=0; i<n; ++i) {
                scanf("%d", &k);
                printf("%d\n", P[k-1]);
        }
        return 0;
}
```

Exercise 1.20

Analysis: All you need to care is how to read input. This problem tests your ability of dealing with strings. The sample code is shown below.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
        char str[30];
        int T, i, j;

        scanf("%d\n", &T);
        for (i = 0; i < T; i++) {
                scanf("%s", str);
                for (j = 0; j < strlen(str); j++) {
                        if (str[j] < '0' || str[j] > '9') putchar(str[j]);
                }
                putchar('\n');
        }
        return 0;
}
```