Software Mathematics HW #3

202033762
장민호

P. 214

7. first, declare an integer type array that we can store a list of n integers.

Second, find the index of last entered number

Third, using 'for' and 'if', find if there is even integer. like this↓

```
for( index; index ≥ 0; index --) {
    if( arr[index] % 2 = 0) {
        return index;
    }
}
return 0;
```

p. 215

36. sort 6, 2, 3, 1, 5, 4

1. 6 2 3 1 5 4

2 6 3 1 5 4

2 3 6 1 5 4

2 3 1 6 5 4

2 3 1 5 6 4

2 3 1 5 4 6

2. 2 3 1 5 4 6

2 1 3 5 4 6

2 1 3 4 5 6

3. 2 1 3 4 5 6

1 2 3 4 5 6

P. 228

2. Determine whether each of these functions is $O(x^2)$.

a) $f(x) = 17x + 11$ b) $f(x) = x^2 + 1000$

c) $f(x) = x \log x$ d) $f(x) = x^4/2$

e) $f(x) = 2^x$ f) $f(x) = \lfloor x \rfloor \times \lceil x \rceil$

$f(x) = O(g(x))$, if there constants C, k, $|f(x)| \le C|g(x)|$ when $x > k$.

In this case, a) $f(x) = 17x + 11$, $|f(x)| \le 18|x|$ when $x > 17$. so a) $= O(x)$

b) $f(x) = x^2 + 1000$, $|f(x)| \le 2|x^2|$ when $x > 10\sqrt{10}$. b) $= O(x^2)$

⋮

f) $f(x) = \lfloor x \rfloor \times \lceil x \rceil \begin{cases} x \text{ has decimal part: } f(x) = x(x+1) = x^2 + x. \\ x \text{ is integer: } f(x) = x^2 \end{cases}$

whether x has decimal part or not, $|f(x)| \le 2|x^2|$ when $x > 1$. So f) $= O(x^2)$

P 229

30. Show that each of these pairs of functions are of the same order.

a) $3x+7$, x.

let $f(x) = 3x+7$, $h(x) = x$.

Let's find $g(x)$ that satisfy $|f(x)| \le C_1 |g(x)|$ & $|h(x)| \le C_2 |g(x)|$ when $x > k$,

$f(x)$: $|3x+7| \le 4|x|$, $x > 7$

$h(x)$: $|x| \le |x|$, for all x.

so $g(x) = x$ then $f(x) = O(x)$, $h(x) = O(x)$

b) $2x^2 + x - 7$, x^2

as we do first, let $f(x) = 2x^2 + x - 7$, $h(x) = x^2$

$f(x)$: $|2x^2 + x - 7| \le 3|x^2|$ when x satisfies $x^2 - x + 7 \ge 0$ ... for all x

$h(x)$: $|x^2| \le |x^2|$ for all x.

so, $g(x) = x^2$ then $f(x) = O(x^2)$, $h(x) = O(x^2)$

c) $\lfloor x + 1/2 \rfloor$, x

$f(x) = \lfloor x + \frac{1}{2} \rfloor$, $h(x) = x$.

$\lfloor x + \frac{1}{2} \rfloor < \begin{matrix} x+1 \\ x. \end{matrix} \rightarrow$ $|x+1| \leq 2|x|$ when $x > 1$

$|x| \leq |x|$ for all x

$\llcorner$ same as $h(x)$

So, $g(x) = x$, then $f(x) = O(x)$, $h(x) = O(x)$

d) $\log(x^2+1)$, $\log_2 x$

$f(x) = \log(x^2+1)$, $h(x) = \log_2 x$

$f(x)$: compare $\log \alpha$, $\log \beta$, it is always true that when $\alpha < \beta$, $\log \alpha < \log \beta$. $(0 < \alpha < \beta)$

so $|x^2+1| \leq 2|x^2|$, $x > 1$

then we can change $\log(x^2+1) \rightarrow \log 2x^2 = 2\log x + \log 2$

$|2\log x + \log 2| \leq 3|\log x|$ when $2 < x$.

$h(x)$: $\log_2 x = \dfrac{\log x}{\log 2} = \dfrac{1}{\log 2} \cdot \log x$ so $\left|\dfrac{1}{\log 2} \cdot \log x\right| \leq |\log x|$, $x > 0$.

So, $f(x) = O(\log x)$, $h(x) = O(\log x)$

e) $\log_{10} x$, $\log_2 x$

$f(x) = \log_{10} x = \log x$, $h(x) = \log_2 x$.

$f(x)$: $\log x \leq \log x$, $x > 0$

$h(x)$: $\log_2 x = \frac{1}{\log 2} \cdot \log x$

$$\frac{1}{\log 2} \cdot \log x \leq \log x, \quad x > 0$$

so $f(x) = O(\log x)$, $h(x) = O(\log x)$

P.241 972

2. Give a big-O estimate for the number additions used in this segment of an algorithm

$t := 0$

for $i := 1$ to n

  for $j := 1$ to n

    $t := t + i + j$

let's suppose t_i is the sum of t when i is fixed value

when $i=1$

$$t_1 = \underbrace{(0+1+1) + (1+2) + (1+3) + \cdots + (1+n)}_{n}$$

$$= n \times 1 + (1+2+\cdots+n) + t_0$$

$$= n + \frac{n(n+1)}{2}$$

$$= \frac{1}{2}(n^2+3n)$$

when $i=2$

$$t_2 = n \times 2 + (1+\cdots+n) + t_1$$

$$= 2n + \frac{n(n+1)}{2} + \frac{1}{2}(n^2+3n) \quad = \frac{1}{2}(2n^2+8n)$$

when $i=3$

$$t_3 = n \times 3 + (1+\cdots+n) + t_2$$

$$= 3n + \frac{1}{2}(n^2+n) + \frac{1}{2}(2n^2+8n)$$

$$= \frac{1}{2}(3n^2+15n)$$

So, $t_i = \frac{1}{2}(n^2+(2i+1)n) + t_{i-1}$

$$t_{i-1} = \frac{1}{2}(n^2+(2(i-1)+1)n) + t_{i-2}$$

$$\vdots$$

$$t_0 = 0$$

next $\longrightarrow$

$$t_i = \frac{1}{2}(n^2 + (2i+1)n) + t_{i-1}$$

$$t_{i-1} = \frac{1}{2}(n^2 + (2(i-1)+1)n) + t_{i-2}$$

$$\vdots$$

$$t_0 = 0$$

$$t_i = \cancel{t_0} + t_1 + \cdots + t_{i-1} + \frac{1}{2}(n^2 + (2i+1)n)$$

$$= \frac{1}{2}\left(n^2 \cdot (i-1) + \left(2 \cdot \frac{i(i-1)}{2} + i\right)n\right) + \frac{1}{2}(n^2 + (2n+1)n)$$

$t_n = C_1 n^3 + \alpha$, (C_1 is constant)

so , $O(n^3)$

P. 242

7. Suppose that an element is known to be among the first four elements in a list of 32 elements. Would a linear search or a binary search locate this element more rapidly?

↳ number means the number of searches.

When we use linear search, the best case is 1, worst case is 4.

In binary search, the best case is 4, worst case is 5

Compare worst case. linear search would locate this element more rapidly!