13/2

Loop Unrolling

For loop

1)
```
for(i=1; i<=n; i++)
    printf("crescent")
```
complexity: $O(n)$

2)
```
for (i=1; i<=n; i++)          ——> O(n)
    for (j=1; j<=n ; j++)     ——> O(n)
        printf ("crescent");
```
complexity: $O(n^2)$ ← $\boxed{n \times n = n^2}$

3)
```
for (i=1; i<=n; i++)          —— O(n)
    for (j=1; j<=n; j++)      —— O(n)
        Printf("crescent");
printf("I am Amsa");          —— O(1)
```
Total time complexity

$O(n) \times O(n) + O(1)$

$O(n^2) + O(1)$ → Constants are ignored

complexity = $O(n^2)$ //

4) for $(i=1;\ \underbrace{i^2<n}_{}\ ;i{+}{+})$ $\rightarrow i \le \sqrt{n}$

printf("crescent");

Time complexity $= O(\sqrt{n})$

5) for $(i=1;\ i<=\frac{n}{2};\ i{+}{+})$ $\longrightarrow \frac{n}{2}$ times

for $(j=1;\ j^2 \le n;\ j{+}{+})$ $\longrightarrow \sqrt{n}$ times

Printf("crescent");

time complexity $= O\left(\frac{n}{2} \times \sqrt{n}\right)$

$= O\left(\frac{n\sqrt{n}}{2}\right) = O(n\sqrt{n})$

6) for $(i=1;\ i<=\frac{n}{2};\ i{+}{+})$ $\longrightarrow \frac{n}{2}$ times

for $(j=1;\ j^2 \le n;\ j{+}{+})$ $\longrightarrow \sqrt{n}$ times

for $(k=\frac{n}{2};\ k \le n;\ k{+}{+})$ $\rightarrow \frac{n}{2}$ times

Printf("crescent");

$$\text{time Complexity} = O\left(\frac{n}{2} \times \sqrt{n} \times \frac{n}{2}\right)$$

$$= O\left(\frac{n^2}{4} \times \sqrt{n}\right)$$

$$= O\left(n^2 \times \sqrt{n}\right)$$

$$\text{Complexity} = O\left(n^2\sqrt{n}\right)$$

7) for(i=1; i<=n; i=i*2)

Printf(" crescent)

$i=1 \ \dots \ 2^0$

$i=2 \quad 2^1$

$i=4 \quad 2^2$

$i=8 \quad 2^3$

$\vdots$

$i=k \quad 2^k$

$$2^k = n$$

$$\log 2^k = \log n$$

$$k \log_2 = \log_n$$

$$\boxed{k = \log_2 n}$$

8) for(i=1; i<=n; i=i*3)

printf("crescent")

$i=1$ $\quad 3^0$

$i=3$ $\quad 3^1$

$i=9$ $\quad 3^2$

$i=27$ $\quad 3^3$

$\vdots$ $\quad \vdots$

$i=k$ $\quad k$

$$3^k = n$$

$$\log_3{}^k = \log_n$$

$$k \log 3 = \log n$$

$$k = \log_3 n$$

9) for(i=1; i<=$\frac{n}{2}$; i++) $\longrightarrow$ $\frac{n}{2}$ times

for (j=1; j<=n; j=j*4) $\longrightarrow$ $\log_4 n$ times

for(k=1; k^2 <=n; k++) $\rightarrow$ $\sqrt{n}$ times

Printf("cresent");

Total time complexity $= O\left(\frac{n}{2} * \log_4 n * \sqrt{n}\right)$

complexity $= O\left(\sqrt{n} * \log_4 n * n\right)$ //.

10) for (i=1; i<=$\frac{n}{2}$; i++) $\longrightarrow$ $\frac{n}{2}$ times

for (j=1; j^2 <=n; j++) $\longrightarrow$ $\sqrt{n}$ times.

for(k=1; k <=500; k++) $\longrightarrow$ 500 times

Printf ("crescent");

Total time Complexity $= O\left(\frac{n}{\cancel{2}} * \sqrt{n} * \overset{250}{\cancel{500}}\right)$

$= O(n * \sqrt{n} * 250)$

ignore the Constants

Complexity $= O(n * \sqrt{n})$

Loop Unrolling

6 m

11. for(i=1; i<=n; i++)
    for(j=1; j<=i; j++)
        for(k=1; k<=100; k++)
            printf("crescent");

i=1   i=2 .... i=n

j=1   j=2 .... j=n

k=100   k=100 .... k=100

[1×100] times   [j=1 → 100 times, j=2 → 100 times] 2×100 times   [j=1 → 100 times, j=2 → 100 times, ⋮, j≤i → i×100 times] n*100 times.

Total complexity =

$(1\times100) + (2\times100) + \dots\dots (n\times100)$ times

$= 100(1+2+3+\dots n)$

$= \cancel{100}^{50}\left(\frac{n(n+1)}{\cancel{2}}\right)$

$= 50(n(n+1))$ [ignore the constants)

Complexity = (n^2+n)

Loop Unrolling

12.
```
for(i=1; i<=n; i++)
    for(j=1; j<=i^2 ; j++)
        for(k=1; k<=n/2 ;k++)
            Printf("crescent");
```

$i=1 \quad i=2 \quad i=3 \; \ldots\ldots \; i=n$ times

$j=1^2 \quad j=2^2 \quad j=3^2 \; \ldots\ldots \; j=n^2$ times

$k=\frac{n}{2} \quad k=\frac{n}{2} \quad k=\frac{n}{2} \; \ldots\ldots \; k=\frac{n}{2}$ times

$(1^2\times\frac{n}{2}\text{ times}) \quad (2^2\times\frac{n}{2}\text{ times}) \quad (3^2\times\frac{n}{2}\text{ times}) \; \ldots\ldots \; (n^2\times\frac{n}{2}\text{ times})$

Total time Complexity =

$$= \left[1^2\times\frac{n}{2} + 2^2\times\frac{n}{2} + 3^2\times\frac{n}{2} + \ldots\ldots n^2\times\frac{n}{2}\text{ times}\right]$$

$$= \frac{n}{2}\left[1^2+2^2+3^2+\ldots\ldots n^2\right]$$

$$= \frac{n}{2}\left[\frac{n(n+1)(2n+1)}{6}\right]$$

(ignore the Constants)

$$= n^4$$

Total Complexity $= O(n^4)$

13) for(i = $\frac{n}{2}$; i<=n; i++) → $n/2$

for(j=1; j<=n; j=j*2 → $\log_2 n$

for(k=1; k<=n; k=k*2 → $\log_2 n$

printf("crescent");

? ⇒ $\log_2 n * \log_2 n * n/2$

$\Rightarrow O(n(\log_2 n)^2)$

14) while (n>1)

{ n = n/2

}

let n=20

20

10

5

2

1

$\Rightarrow \lfloor \log_2 20 \rfloor$

$= O(\log_2 n)$

15) while (n>1)

n = n/5

$\Rightarrow O(\log_5 n)$

16) for (i = $\frac{n}{2}$; i<=n; i++) → $\frac{n}{2}$

for (j=1; j<=$\frac{n}{2}$; j++) → $\frac{n}{2}$

for(k=1; k<=n; k=k*2) → $\log_2 n$

Printf("crescent")

No loop unrolling

loop is independent.

$\frac{n}{2} * \frac{n}{2} * \log_2 n$

$\Rightarrow O(n^2 \log_2 n)$

(17) for (i=1 ; i<=n ; i++)

for (j=1 ; j<= n ; j=j+i)

printf (" crescent")

$i=1$ $i=2$ ...... $i=3$ ... $i=n$

$j=1$ to n $j=1,3,5$ ...... $j=1,4,7$ ...... $\frac{n}{3}$

n times $\frac{n}{2}$ times $\frac{n}{3}$ times

$$\therefore\ n + \frac{n}{2} + \frac{n}{3} + \cdots \frac{n}{n}$$

$$\Rightarrow n\left(1 + \frac{1}{2} + \frac{1}{3} + \cdots \frac{1}{n}\right)$$

$$\Rightarrow n(\log n) \Rightarrow O(n\log n)$$

Loop Unrolling (dependent loop)

18    $n = 2^{2^k}$

```
for ( i=1; i<=n; i++)
{
    j=2;
    while(j<=n)
    {
        j=j^2
        printf ("crescent");
    }
}
```

$k=1$

$n = 2^2 = 4$

$j=2$ $j=4$ $j=16$

$n \times 2$ times

$k=2$

$n = 2^{2^2} = 2^4 = 16$

$j=2$ $j=4$ .. $j=16$

$n \times 3$ times.

$k=3$

$n = 2^{2^3} = 2^8 = 256$

$j=2$ $j=4$ .. $j=16$

$n \times 4$ times

$n \times (k+1)$ times

$n = 2^{2^k}$

Take log on both sides

$\log_2 n = 2^k$

$\log\log n = k$

$n(\log\log n + 1) = O(n\log\log n)$