Perform analysis

Statments not Considerd as step : 1) Variable decleration 2) Method headers 3) الأقواس

1) Assigment operator “ int A = 1 , int B = 2 ; ” : 1 لأن ال Statment تنتهي بـ ؛ نحسبها 1

2) Method Call “ System.o.p (“ ”); ” : 1

3) Return Statments “ return 22 ; ” : 1

4) increment , Decrement “ M++ ” : 1

5) Comparision “ IF (A < 5) ” : 1

Ex.

```
IF (A < 5) {
      System.out.println("A<5");
   } else
   IF (A < 7) {
      System.out.println("A<7");
      System.out.println("!!! ");
   } else
    System.out.println("A>7");
```

Since A is not specified, we consider the worst case

So total = 4

Big-O = O(1)

What is the Best Case ?! 2

6) Loops :

1) عدد الدورات محدد ↓ Total = number

2) "n" عدد الدورات مجهول ↓ Total = n يكون فيه

a. Single Loop "العداد بدون قسمة او ضرب"

1) number ≤ n

$\# \text{Iteration} = \frac{\text{Max-min}}{\text{مقدار الزيادة}} + 1$

$\# \text{check} = \text{Iteration} + 1$

2) number > n

$\# \text{Iteration} = \frac{\text{Max-min}}{\text{مقدار الزيادة}}$

$\# \text{check} = \text{Iteration} + 1$ ← Loop header دائمًا نزيد 1 على ال

`for(int i=1 ; i≤n ; i++){`	$n-1+1+1=n+1$
`    System.out.println(i);`	$n-1+1=n$
`}`	

Total= 2n+1

Big-O = O(n)

`int i=1;`	1
`while (i<10){`	$10-1+1=10$
`  i++; }`	$10-1=9$

Total = 20

Big-O = O(1)

`for(int i=1 ; i≤n ; i+=2){`	$\frac{n-1+1}{2}+1=\frac{n}{2}+1$
`    System.out.println(i);}`	$\frac{n-1+1}{2}=\frac{n}{2}$

$\text{Total} = \frac{2n+1}{2}$

Big-O = O(n)

`int i=1;`	1
`do{`	
` i++;`	$n-1+1+1=n+1$
`}while(i≤n);`	$n-1+1+1=n+1$

do-while special!

Total = 2n+2

Big-O = O(n)

`int i=1;`	1
`while (i>10){ }`	1

Total = 2

Big-O = O(1)

b. Single Loop “العداد فيه قسمة او ضرب”

1) number ≤ n

Iteration = Log(max) - Log(min) +1

Check = Iteration + 1

```
for(int i=1 ; i≤n ; i/=2){
    System.out.println(i);
}
```

$\log(n) - \underbrace{\log(1)}_{=0} + 1 + 1$

$\log(n) - \log(1) + 1$

Total = 2logn+3

Big-O = O(logn)

2) number > n

Iteration = Log(max) - Log(min)

Check = Iteration + 1

```
for(int i=2 ; i<16 ; i*=2){
    System.out.println(i);}
```

$\underbrace{\log(16)}_{=4} - \underbrace{\log(2)}_{=1} + 1$

$\log(16) - \log(2)$

Total = 7

Big-O = O(1)

C. Nested Loop "ماتعتمد على بعض"

1) **Outer Loop:** نفس الـ Single loop

2) **inner Loop:** 1 نحسبها عادى زي الـ Single 2 بعدين اضرب النتيجة بـ #Iterations حقت الـ Outer

```
for(int i=1 ; i<=n ; i++){                      n-1+1+1 = n+1
    System.out.println("you");                  n-1+1 = n
    for (int j =0 ; j < n ; j ++ ){             n(n+1)   } inner loop
        system.out.println ("can");             n(n)     }
    }
    System.out.println("do it!").               n
}
```

$$\text{Total} = 4n + 2n^2 + 1$$

$$\text{Big-O} = O(n^2)$$

d. Nested Loop "تعتمد على بعض"

1) Outer Loop: نفس ال Single loop

2) inner Loop: $\# \text{ Iteration} = \dfrac{\# \text{ outer Loop Iterations } (\overset{\overset{\text{outerloop}}{\uparrow}}{\text{Max}} + \overset{\overset{\text{outerloop}}{\uparrow}}{\text{Min}})}{2}$

$\# \text{ check} = \dfrac{\# \text{ outer Loop Iterations } (\overset{\overset{\text{outerloop}}{\uparrow}}{\text{Max}}+1 + \overset{\overset{\text{outerloop}}{\uparrow}}{\text{Min}}+1)}{2}$

```
for(int i=1 ; i<n ; i++){                       n-1+1=n

   System.out.println("you");                   n-1

   for (int j = i ; j<=n ; j ++ ){               (n-1)(n+4)
                                                     2          } inner Loop
      System.out.println ("can");               (n-1)(n+2)
                                                     2
   }

   System.out.println("So it! ").               n-1

}
```

Big-O = $O(n^2)$

How to specify the big-O ?

1) Drop All constants

2) نختار أكبر دالة موجودة

إذا الأس أكبر من　　any constant

$Constant < \log n < n < n \log n < n^{2 \dots \infty} < 2^{n} < n! < n^{n}$

$\log \log n < \log n$

$\log n < \sqrt[2..\infty]{n}$　　اللوق أصغر من الجذر دائمًا

- $C =$ قيمتها متغيرة لكن نعتمد مجموع المعاملات، نتجاهل السالب
- $n_0 =$ متغيرة لكن نعتمد 1 وإذا ال Big-O فيه log تصير 2

$T(n) = n^2 \log n + n^2$

$Big\text{-}O = O(n^2 \log n)$

$C = 2 \; , \; n_0 = 2$

$T(n) = n^n \log n + n!$

$Big\text{-}O = O(n!)$

$C = 2 \; , \; n_0 = 1$

$T(n) = n^2 \log n + n^n$

$Big\text{-}O = O(n^n)$

$C = 2 \; , \; n_0 = 1$

$T(n) = n^n + 2^n$

$Big\text{-}O = O(n^n)$

$C = 2 \; , \; n_0 = 1$

1 semester 2022/2023

دعواتكم ♡