Loop time complexity practice 9/6/2024

I. Time complexity of loops

Give the time complexity for the pieces of code below. Use either O or Theta (Θ) for time complexity.

The answer should be in the format covered in class.

Closed form for a summation is the answer formula for it. E.g. *1+2+3+4+…+N* has closed form *N(N+1)/2* .

Sample table for answers

A. Normal

0. For each of the questions below say which letters can show up in the final time complexity. You should answer this BEFORE you solve them. E.g. i,N,k,M,t,S etc

1.

```
for(i=1; i<=3N; i=i+3)
     for(k=1; k<=M; k=k*6)
          printf("C");
```

2.

```
for(i=0; i<=(N+1); i=i+S)
     for(t=1; t<=M; t=t+3)
          printf("B");
```

3.

```
for(i=0; i<=N; i=i+7)
```

```
     for(k=1; k<=i; k=k+2)
          printf("B");
```

4.

```
for(i=N; i>=1; i=i/2)
     for(k=0; k<=M; k=k+S)
          printf("B");
```

5.

```
for(i=N; i>=0; i=i-5)
     for(k=N; k>=1; k=k/3)
          printf("B");
```

6. Good, with function calls

```
// int mistery(int len, int v); has Θ(len²)
for(i=N; i>=0; i--){
     int res = mistery(N, i);
     for(k=N; k>=1; k=k-1)
          printf("B");

}
```

7. Good.

```
// int mistery(int N, int v); has Θ(N²)
for(i=N; i>=0; i--){
     int res = mistery(i, i);
     for(k=N; k>=1; k=k-1)
          printf("B");

}
```

8. Good

```
for (i = 101; i<=(100+N); i++)
   for (k=1; k<=i; k = k+1)
      printf("B ");
```

9. Good

```
for (i=1; i<=N; i=i*2)
   for (k=1; k<=2*i; k=k+1)
      for (t=0; t<7; t=t+1)
         printf("A");
```

10.

```
for(t=1; t<=N; t=t*3)
     for(i=10; i<=N; i=i+2)
          for(k=N; k>=1; k=k-5)
               printf("G");
```

11.

```
// *** good example
for(i=1; i<=N; i=i+1)
     for(k=1; k<=i; k=k+k)
          for(t=1; t<=S; t++)
               printf("G");
```

12.

```
for (i=1; i<=N; i=i+1) {
   for (k=1; k<=i; k++) {
      printf("B");
      for (t=1; t<=k; t++)
         printf("C");
   }  // end for-k
…
} // end for-i
```

13. good.

```
// Assume void do_smth(int N);has Θ(N)
for (i=1; i<=N; i=i+1) {
   for (k=1; k<=i; k++) {
      printf("B");
      for (t=1; t<=k; t++)
         do_smth(k);
   }  // end for-k
…
} // end for-i
```

14.

```
// example from Dr. Weems
for (i=1; i<=N-1; i++) {
  …
  for (k=i+1; k<=N; k++)  {
    temp=ab[ell][k];
    ab[ell][k]=ab[i][k];
    ab[i][k]=temp;
    for (t=i+1; t<=N; t++)
      ab[t][k] -= ab[t][i]*ab[i][k];
    }// end for-k
   …
} // end for-i
```

15.

```
for (i=1; i<=N; i=i+i) {
   for (k=1; k<=N; k++)  {
      printf("B");
      for (t=1; t<=N; t++)
         printf("C");
   } // end for-k
   …
} // end for-i
```

16. 3-level; Math manipulation required

```
for (i=1; i<=N; i=i*2) {
   for (k=0; k<=i; k=k+3)  {
      printf("B");
      for (t=1; t<=k; t++)
         printf("C");
   } end for-k
      …
}  // end for-i
```

17. Non-trivial/Hard and additional questions

```
for(i=N; i>=1; i--)
     for(k=i; k>=1; k=k/2)
          for(t=i; t<=N; t=t+2)
               printf("G");
```

1) Write only the summation for T(N). You do NOT need to find the closed form or Theta (Θ).

2) Can you give a CLOSE lower bound and an upper bound for this time complexity? Saying that this code takes more than lgN to run and less than N^3. Is NOT CLOSE enough.

B. Non-standard loop: loop variable updates in body of loop as well.

```
for(i=1; i<=N; i=i+3){
   printf("X");
   i=i+5;
   printf("Y");
}
```

C. i (outer) is updated based on k (inner)

Compute the time complexity (Θ) for all the code samples in this section.

1) Given the code below, assume M and N were declared and prior to execution of this code.

```
for(i=0, k=0; i<=N; i=i+k){
     printf("%d\n", k);
     for(k=0; k<M; k=k+1)
          printf("G");
```

What will this program print for M = 3 and N = 14?

True/False: if we remove k=0 from the for(k…) loop, it will **not** change the **TC** of the entire code.
True/False: if we remove k=0 from the for(i…) loop, it will **not** change the **TC** of the entire code.

2)

```
for(i=0, k=0; i<=N; i=i+k)
      for(k=0; k<M; k=k+1)
            for(t=0; t<=k; t=t+1)
                  printf("G");
```

3)

```
for(i=0; i<=N; i=i+k)
      for(k=0; k<S; k=k+1)
            printf("B");
```

4)

```
i = 0;
while (i<=N){
      for(t=0, k=1; k<N; t=t+1, k=2*k)
            printf("G");
      i=i+t;
}
```

II. Generate code of certain time complexity

A. Give a piece of code with nested loops that has time complexity NlgN.

III. Exact count of loop iterations – Extra. NOT required

For each piece of code below write the function that gives the exact count of loop iterations.

A. Variations

```
for(k=7; k<=M; k=k+6)
   printf("A");

for(k=7; k<M; k=k+6)  // same as above, but k<M
   printf("A");

for(i=7; i<=M; i=i*6)
      printf("A");

for(i=1; i<=N; i=i+i)
      printf("A");

for(i=1; i<=2N; i=i+2)
      printf("A");
```