TC - conditionals (if)

Saturday, January 22, 2022 7:28 PM

Examples

Assume that in the code pieces below, each function call has the time complexity shown next to it. Give the Worst, Best, and possibly Average time complexity. Also give the general time complexity (what you would report).

- Example 1
  ```
  if (valid(val)){   // O(1)
      foo(arr, N, val) ; // O(N)
  }
  else {
      printf("Error");   // O(1)
  }
  ```
  Worst: O(____) , Best: O(____), Average: O(____), General TC: O(____)

- Example 2 (no false branch)
  ```
  if (valid(val)){   // O(1)
      foo(arr, N, val) ; // O(N)
  }
  ```
  How many paths are possible when this code executes?
  Worst: O(____) , Best: O(____), Average: O(____), General TC: O(____)

- Example 3
  ```
  if (validBig(val,N)){   // O(N)
      foo(arr, N, val) ; // O(N)
  }
  else {
      printf("Error");   // O(1)
  }
  ```
  Worst: O(____) , Best: O(____), Average: O(____), General TC: O(____)

- Example 4
  ```
  if (validBig(N,val)){   // O(N)
      fooSmall(N, val) ; // O(1)
  }
  else {
      bar(N);   // O(N^2)
  }
  ```
  Worst: O(____) , Best: O(____), Average: O(____), General TC: O(____)

- Example 5
  Assume array A is sorted in increasing order and has N elements.
  ```
  k=N-1;
  while ((k>=0)&&(A[k]>val)){
      k--;
  }
  //if ((k<0)||(A[k]!=val) printf("not found");
  //else printf("found val in A");
  ```
  Worst: O(____) , Best: O(____), Average: O(____), General TC: O(____)

General guidelines

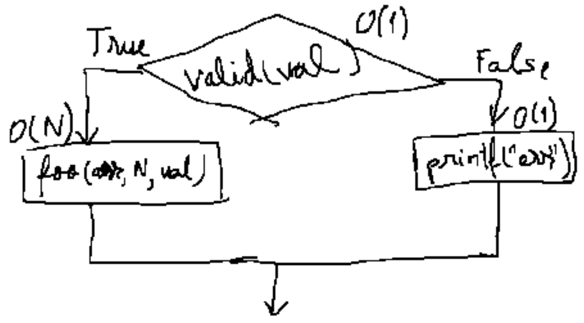
1. When giving the general TC, we give the WORST case.
2. If possible, analyze and give best and average cases as well.
3. When analyzing average case, assume uniform distribution (each event/case is equally likely to happen).

Solutions:

Example 1

Compute TC for the code below. Assume the given TC for each line is correct.

```
if (valid(val)){   // O(1)
    foo(arr, N, val) ; // O(N)
}
else {
    printf("Error");   // O(1)
}
```



Worst: O(**N**) , Best: O(**1**), Average: O(**N**), General TC: O(**N**)
Worst case: O(N)   (this will be used when reporting TC for this code)
Best case: O(1)
Average case: (O(N)+O(1))/2 = O(N)   (We have 2 events: go on the TRUE branch or on the FALSE branch. If they both have equal probability to happen, the average case is (O(N)+O(1))/2 = O(N))
For the general time complexity we say O(N) (we give the worst case)

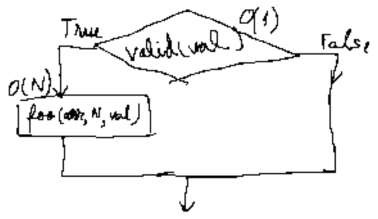
Example 2 (no FALSE branch)

Assume the same example as above, but with the else branch removed.
Compute TC for the code below. Assume the given TC for each line is correct.

```
if (valid(val)){   // O(1)
    foo(arr, N, val) ; // O(N)
}
```

How many paths are possible when this code executes?
Two paths. Even if I do not see an else branch, if the `valid(val)` is FALSE, we take a different path and we have TC: O(1)   (the TC of `valid(val)`)
Worst: O(**N**) , Best: O(**1**), Average: O(**N**), General TC: O(**N**)

Best case: O(1)
Average case: (O(N)+O(1))/2 = O(N)   (same as in the example above)



Example 3

Compute TC for the code below. Assume the given TC for each line is correct.

```
if (validBig(val,N)){   // O(N)
    foo(arr, N, val) ; // O(N)
}
else {
    printf("Error");   // O(1)
}
```

Worst: O(**N**) , Best: O(**N**), Average: O(**N**), General TC: O(**N**)
Worst case: O(N)   (the validBig(val,N) and foo(arr, N, val) are executed SEQUENTIALLY and their TC will be added: O(N)+O(N) = O(N))
Best case: O(N)   (from the TC of `validBig(val,N)`)
(Average case: O(N))

Example 4

Compute TC for the code below. Assume the given TC for each line is correct.

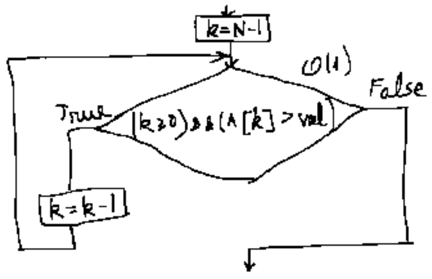
```
if (validBig(N,val)){   // O(N)
    fooSmall(N, val) ; // O(1)
}
else {
    bar(N);   // O(N²)
}
```

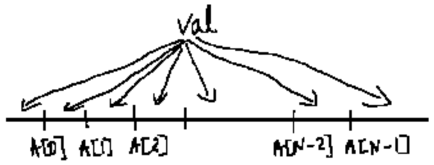
Worst: O(**N^2**) , Best: O(**N**), Average: O(**N^2**), General TC: O(**N^2**)
Worst case: O(N^2)   (from O(N) + O(N^2) = O(N)
Best case: O(N)   (from the TC of `validBig(N, val)`)
Average case: O(N^2)

Example 5

Assume array A is sorted in increasing order and has N elements.

```
k=N-1;
while ((k>=0)&&(A[k]>val)){
    k--;
}
//if ((k<0)||(A[k]!=val) printf("not found");
//else printf("found val in A");
```





Worst: O(**N**) , Best: O(**1**), Average: O(**N**), General TC: O(**N**)

Assume uniform distribution. Each event is how many times the loop executes (that is how many times the k--; instruction executes). k-- can execute 0, 1, 2, 3, …, N times, thus we have N+1 possible events and each one of them has equal probability.

$$Average = \frac{sum\ of\ value\ of\ each\ event}{number\ of\ events} = \frac{0+1+2+3+\cdots+N}{N+1} = \frac{\frac{N(N+1)}{2}}{N+1} = \frac{N}{2}$$

While we drop the constants for O() thus both worst and average case are O(N), you can see that in the average case it takes half the time it does in the worst case.