DS 1^{st} homework

58121102 Jiang Yuchu

September 19 2022

3. Determine the frequency counts for all statements in the following two program segments:

(a)

```
1  for(i=1;i<=n;i++) // n+1 times
2    for(j=1;j<=i;j++) // i times for every i -> (1+2+3+4...n)
3      for(k=1;k<=j;k++) // j times for every i,j
4        x++;              // -> (1+(1+2)+(1+2+3).....(1+2+3...n))
```

Answer:

Line	Frequency Count
1	n+1
2	$\frac{(1+n)n}{2}$
3	$\frac{n(n+1)(n+2)}{6}$
4	$\frac{n(n+1)(n+2)}{6}$

(b)

```
1  i=1; // 1 times
2  while(i<=n) { // n+1 times
3    x++; // n times
4    i++; // n times
5  }
```

Answer:

Line	Frequency Count
1	1
2	n+1
3	n
4	n

4. (a) Introduce statements to increment *count* at all appropriate points in Program 1.32.

```
1  void D(int* x, int n) {
2    int i = 1;
3    do {
4      x[i] += 2;
5      i += 2;
6    } while(i <= n);
7    i = 1;
8    while(i <= (n / 2)){
9      x[i] += x[i+1];
10     i++;
11   }
12 }
```

Program 1.32: Example Program

Answer:

```
1  void D(int* x, int n) {
2    int i = 1, count = 1;
3    do {
4      x[i] += 2;
5      i += 2;
6      count += 3;
7    } while(i <= n);
8    i = 1, ++count;
9    while(i <= (n / 2)){
10     x[i] += x[i+1];
11     i++;
12     count += 3;
13   }
14   count++;
15 }
```

(b) Simplify the resulting program by eliminating statements. The simplified program should compute the same value for *count* as computed by the program of (a).

Answer:

```
1  void D(int* x, int n) {
2    int count = 0;
3    for (int i = 1; i == 1 || i <= n; i += 2)
4      count += 3;
5    for (int i = 1; i == 1 || i <= (n / 2); i++)
6      count += 3;
7    count += 3; // for isolated statements
8  }
```

(c) What is the exact value of *count* when the program terminates? You may assume that the initial value of *count* is 0.

Answer:

$count = 3n + 3$

(d) Obtain the step count for Program 1.32 using the frequency method. Clearly show the step count table.

Answer:

Line	S/E	Frequency		Total Steps	
		n is even	n is odd	n is even	n is odd
2	1	1		1	
3	0	$\frac{n}{2}$	$\frac{n+1}{2}$	0	
4	1	$\frac{n}{2}$	$\frac{n+1}{2}$	$\frac{n}{2}$	$\frac{n+1}{2}$
5	1	$\frac{n}{2}$	$\frac{n+1}{2}$	$\frac{n}{2}$	$\frac{n+1}{2}$
6	1	$\frac{n}{2}$	$\frac{n+1}{2}$	$\frac{n}{2}$	$\frac{n+1}{2}$
7	1	1		1	
8	1	$\frac{n}{2}+1$	$\frac{n-1}{2}+1$	$\frac{n}{2}+1$	$\frac{n-1}{2}+1$
9	1	$\frac{n}{2}$	$\frac{n-1}{2}$	$\frac{n}{2}$	$\frac{n-1}{2}$
10	1	$\frac{n}{2}$	$\frac{n-1}{2}$	$\frac{n}{2}$	$\frac{n-1}{2}$
11	0	$\frac{n}{2}$	$\frac{n-1}{2}$	0	

10. Obtain the average run time of function BinarySearch (Program 1.10) Do this for suitable values of n in the range [0,100]. Your report must include a plan for the experiment as well as the measured times. These times are to be provided both in a table and as a graph.

Answer:

Here is what the BinarySearch looks like:

```
1  int BinarySearch(int* a, const int x, const int n) {
2      int left = 0, right = n - 1;
3      while (left <= right) {
4          int middle = (left + right) / 2;
5          if (x < a[middle])
6              right = middle - 1;
7          else if (x > a[middle])
8              left = middle + 1;
9          else
10             return middle;
11     }
12     return -1;
13 }
```

First, search each value from [0, 100] in the range [0, 100]. Second, repeat searching operation for every n for 5×10^7 times. Third, calculate the total running time and average running time. Then, ouput the result into data.csv. Finally, draw the graph of the data by python. Here is cpp code . Here is python code. Here is data table.