Lab 2 Solution

Student Information:

First name: Bo

Last name: Dai

Student ID: 132954173

For each function, write your analysis in the following format.

Factorial Function Analysis:

Step 0: Write code here:

```
unsigned int factorial(unsigned int n) {
  unsigned int temp = 1;
  for (unsigned int i = 2;i <=n;i++) {
          temp = i * temp;
  }
  return temp;
}
```

Step 1: Establish variables and functions (mathematical ones) and write them down here:

- Let n represent the value we are finding the factorial for
- Let T(n) represent number of operations needed to find n! using the code

Step 2: Count your operations (Write your solution here)

- In the code below there are 3 operations that we do exactly one time

```
  unsigned int temp = 1;
  unsigned int i = 2;
  return temp;
```

- There are also 4 operations that we do each time we go through the for loop that will run n-1 times

```
  i<=n;
  i++;
  temp=i*temp;   (2 ops here)
```

Step 3: Establish the Mathematical Expression for ***T(n)*** (Write the mathematical definition of ***T(n)*** function here)

T(n)=4(n-1)+3

Step 4: Simplify your Equation and write it down here

T(n)=4n-4+3=4n-1

Step 5: State your final result here

Therefore, T(n) is O(n)

Power Function Analysis:

Step 0: Write code here:

```
double power(double base, unsigned int n) {
  double temp = 1.0;
  for (unsigned int i = 1;i <=n;i++) {
          temp = temp*base;
  }
  return temp;
}
```

Step 1: Establish variables and functions (mathematical ones) and write them down here:

- Let n represent the value we are finding the Power for
- Let T(n) represent number of operations needed to find base^n using the code

Step 2: Count your operations (Write your solution here)

- In the code below there are 3 operations that we do exactly one time

```
  double temp = 1.0;
  unsigned int i = 1;
  return temp;
```

- There are also 4 operations that we do each time we go through the for loop that will run n times

```
  i<=n;
  i++;
  temp=temp*base;   (2 ops here)
```

Step 3: Establish the Mathematical Expression for ***T(n)*** (Write the mathematical definition of ***T(n)*** function here)

T(n)=4n+3

Step 4: Simplify your Equation and write it down here

T(n)=4n+3

Step 5: State your final result here

Therefore, T(n) is O(n)

Fibonacci Function Analysis:

Step 0: Write code here:

```
unsigned int fibonacci(unsigned int n) {
  unsigned int f0 = 0;
  unsigned int f1 = 1;
  unsigned int fnext = 0;
  if (n == 0) return 0;
  if (n == 1) return 1; //for f0 and f1
  for (unsigned int i = 0;i < n-1;i++) {        //for n>1
            fnext= f0+f1;
            f0 = f1;
            f1 = fnext;
  }
  return fnext;
}
```

Step 1: Establish variables and functions (mathematical ones) and write them down here:

- Let n represent the value we are finding the Fibonacci for
- Let T(n) represent number of operations needed to find Fn using the code

Step 2: Count your operations (Write your solution here)

- In the code below there are 4 operations that we do exactly one time

```
  unsigned int f0=0;
   unsigned int f1=1;
   unsigned int fnext = 0;
  return 0; Or return 1; Or return fnext;    (if n==0, will run return 0;
if n==1, will run return1, but only 1 return will run for all
situation, so count 1 operation here)
```

- Going next, there are 3 situations need to consider:

- If n==0, there are also 1 extra operation that we do exactly one time before end of program

```
if (n == 0)
```

- If n==1, there are 2 extra operations that we do exactly one time before exit

```
if (n == 0)
if (n == 1)
```

- If n>1, run the loop, there are 3 operations that we do exactly one time

```
if (n == 0)
if (n == 1)
unsigned int i = 0
```

also 7 operations that we do each time we go through the for loop that will run n-1 times

```
i < n-1;   (2 ops here)
i++;
fnext= f0+f1;   (2 ops here)
f0 = f1;
f1 = fnext;
```

Step 3: Establish the Mathematical Expression for ***T(n)*** (Write the mathematical definition of ***T(n)*** function here)

- Based on the above analysis, for special cases when n=0 and n=1:

  If n==0   T(n)=4+1

  If n==1   T(n)=4+2

- For general cases when n>1

  If n>1   T(n)=7(n-1)+4+3

Step 4: Simplify your Equation and write it down here

- Since we are doing the analysis on cost for general cases:

  T(n)=7n-7+4+3=7n

  (ps: for special case： If n==0 T(n)=5;   If n==1 T(n)=6)

Step 5: State your final result here

Therefore, T(n) is O(n)

Submission:

You need to submit a single **PDF** file for this lab. The name of the file must be as follow:

yourLastName_Lab02.pdf

Write the five steps analysis for each function in a text file (e.g. MS Word) and save it as a **PDF** file. Submit the **PDF** file from the Lab 02 link available on the Blackboard.