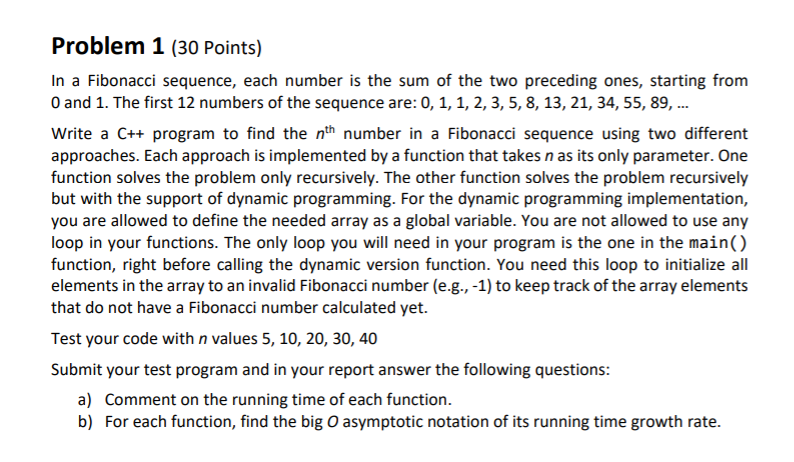
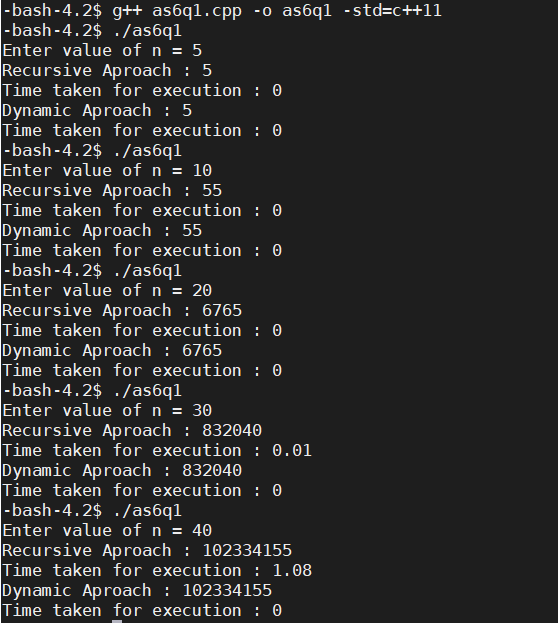
**EECE 7205 : Fundamentals of Computer Engineering**

**Assignment 6**





Q1 a )

Running time of recursive Fibonacci is more than that of dynamic approach because you are calculating Fibonacci(4) you need Fibonacci(3) and Fibonacci(2), Now for Fibonacci(3), you need Fibonacci (2) and Fibonacci (1) but you notice you have calculated Fibonacci(2) while calculating Fibonacci(4) and again calculating it. So, we are solving many sub-problems again and again. In case of dynamic approach, we store these values and reduce the running time. Also it is clearly visible from the output as the value of n increase the recursive approach’s running time goes on increasing.

Q1 b)

Recursive Approach :

T(n) = T(n-1) + T(n-2) + c

= 2T(n-1) + c //from the approximation T(n-1) ~ T(n-2)

= 2\*(2T(n-2) + c) + c

= 4T(n-2) + 3c

= 8T(n-3) + 7c

= 2^k \* T(n - k) + (2^k – 1)\*c

Let's find the value of k for which: n - k = 0

k = n\*T(n) = 2^n \* T(0) + (2^n - 1)\*c

= 2^n \* (1 + c) – c

i.e. T(n) ~ 2^n

Dynamic approach :

T(n) = T(n-1) + c

= T(n-2) + 2c

= T(n-3) + 3c

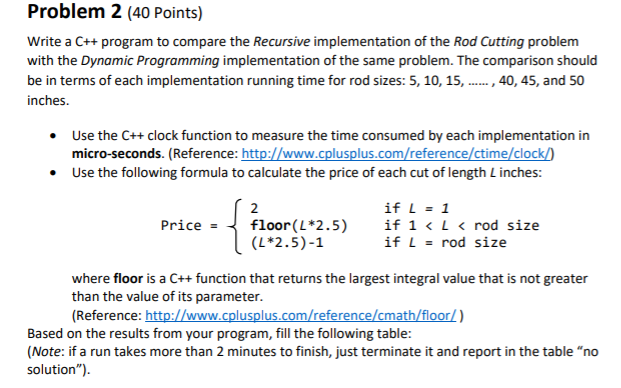
= T(n-k) + kc

Let's find the value of k for which: n - k = 0

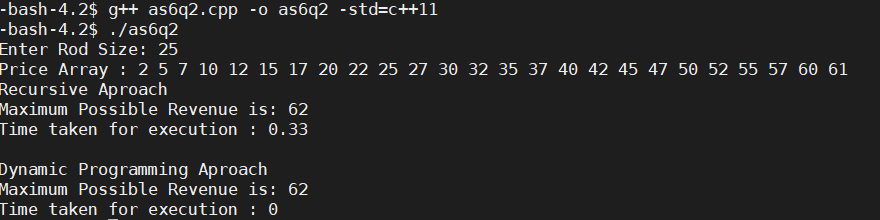
k = n\*T(n) = T(0) + n\*c

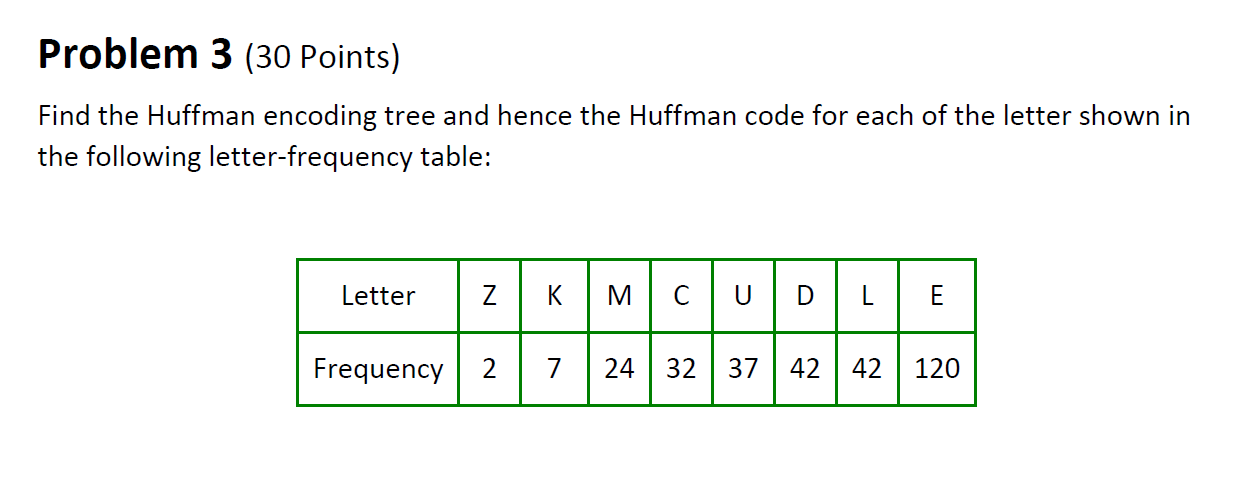
= n\*c + 1

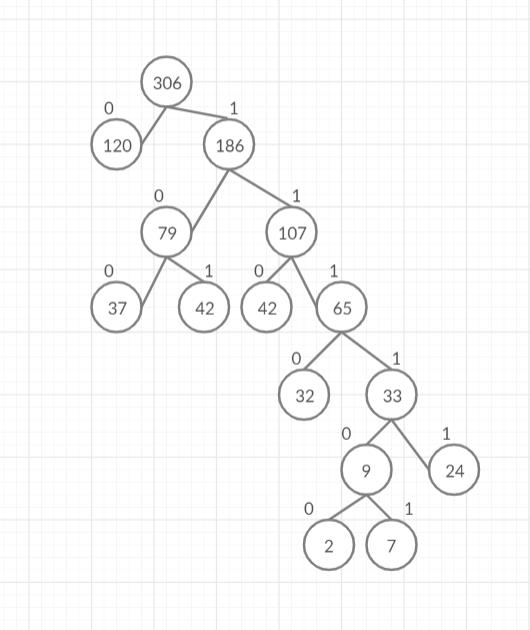
i.e. T(n) ~ n



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rod Size | Recursive  Time | Recursive Time Revenue | Dynamic  Time | Dynamic Time  Revenue |
| 5 | 0 | 12 | 0 | 12 |
| 10 | 0 | 25 | 0 | 25 |
| 15 | 0 | 37 | 0 | 37 |
| 20 | 0 | 50 | 0 | 50 |
| 25 | 0.33 | 62 | 0 | 62 |
| 30 | 9.6399 | 75 | 0 | 75 |
| 35 | >2mins | - | 0 | 87 |
| 40 | >2mins | - | 0 | 100 |
| 45 | >2mins | - | 0 | 112 |







*Huffman Tree*

|  |  |  |
| --- | --- | --- |
| Letter | Frequency | Huffman Code |
| E | 120 | 0 |
| L | 42 | 101 |
| D | 42 | 110 |
| U | 37 | 100 |
| C | 32 | 1110 |
| M | 24 | 11111 |
| K | 7 | 111101 |
| Z | 2 | 111100 |

*Letter Frequency Table*