Marci M. McBride

CSE 122

HW4

1. 5n4+6n2+2n+4 <= 5n4+6 n4+2 n4+4 n4

= 17 n4

O(n4)✔

c= 17

no=1 then 17<= 17 ✔

1. [n\*(n+1)\*(2n+1)] / 6

[(n2+n)\*(2n+1) ]/6

2n3+3n2+n /6 = 2n3+3n3+n3 /6n3

=((6/6) n3)

=n3

O(n3)✔

C = 1

n0 = 1

1. 2n+1

2n\*21 =2 \*2n ✔

O(2n)✔

C = 2

n0 = 0

1. nlogn

2n log2n

2n(log2 + logn)

since were using log base 2

2n(logn)

so if n is doubled the running time is times more

1. Lets calculate n:

|  |  |  |  |
| --- | --- | --- | --- |
| n | logn | nlogn | 2n |
| 10 | 3 | 30 | 103 |
| 103 | 9 | 9x103 | 10300 |
| 106 | 18 | 18x106 | 10300000 |
| 109 | 27 | 27x109 | 10300000000 |
| 1012 | 36 | 36x1012 | 10300000000000 |
| 1015 | 45 | 45x1015 | 10300000000000000 |
| 1018 | 54 | 54x1018 | 10300000000000000000 |
| 1021 | 63 | 63x1021 | 10300000000000000000000 |

Now that I have calculated n, I will calculate how many seconds it will take each algorithm to run on the machine capable of 1012 ops per second.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| n | O(logn) | O(n) | O(nlogn) | O(n2) | O(n3) | O(2n) | O(n!) |
| 10 | 3 x10-12 | 10-11 | 3 x10-11 | 10-10 | 10-9 | 10-9 | 10-6 |
| 103 | 9 x10-12 | 10-9 | 9x10-9 | 10-6 | 10-3 | 10288 | 102555 |
| 106 | 18 x10-12 | 10-6 | 18x10-6 | 1 | 106 | 1010^5.5 | 1010^6.7 |
| 109 | 27 x10-12 | 10-3 | 27x10-3 | 106 | 1015 | 1010^8.5 | 1010^9.9 |
| 1012 | 36 x10-12 | 1 | 36 | 1012 | 1024 | 1010^11.5 | 1010^13.1 |
| 1015 | 45 x10-12 | 103 | 45x103 | 1018 | 1033 | 1010^14.5 | 1010^16.2 |
| 1018 | 54 x10-12 | 106 | 54x106 | 1024 | 1042 | 1010^17.5 | 1010^19.2 |
| 1021 | 63 x10-12 | 109 | 63x109 | 1030 | 1051 | 1010^20.5 | 1010^22.3 |

under a second

in an hour

in a day

in 10 days

in 120 days

in a year

Algorithms with no hope of ever finishing

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| n | O(logn) | O(n) | O(nlogn) | O(n2) | O(n3) | O(2n) | O(n!) |
| 10 | 3 x10-12 | 10-11 | 3 x10-11 | 10-10 | 10-9 | 10-9 | 10-6 |
| 103 | 9 x10-12 | 10-9 | 9x10-9 | 10-6 | 10-3 | 10288 | 102555 |
| 106 | 18 x10-12 | 10-6 | 18x10-6 | 1 | 106 | 1010^5.5 | 1010^6.7 |
| 109 | 27 x10-12 | 10-3 | 27x10-3 | 106 | 1015 | 1010^8.5 | 1010^9.9 |
| 1012 | 36 x10-12 | 1 | 36 | 1012 | 1024 | 1010^11.5 | 1010^13.1 |
| 1015 | 45 x10-12 | 103 | 45x103 | 1018 | 1033 | 1010^14.5 | 1010^16.2 |
| 1018 | 54 x10-12 | 106 | 54x106 | 1024 | 1042 | 1010^17.5 | 1010^19.2 |
| 1021 | 63 x10-12 | 109 | 63x109 | 1030 | 1051 | 1010^20.5 | 1010^22.3 |

1. O(n2)
2. O(n)
3. The second case that uses Horner’s Rule.
4. Best case: O(n2)

Worst case: O(n2)

Therefore the average case is O(n2)

1. Best case: O(n2)

Worst case: O(n2)

Therefore the average case is O(n2)

1. bk = bk-1 / (1+bk-1)

b0 =1

b1 =b1-1/1+b1-1 = 1/1+1 = ½

b2 = b2-1/1+b2-1 =½/1+½ =1/3

b3= b3-1/1+b3-1=(1/3)/1+(1/3) = ¼

b4= b4-1/1+b4-1=(1/4)/1+(1/4) = 1/5

b5= b5-1/1+b5-1=(1/5)/1+(1/5) = 1/6

bn = 1/(n+1)

base case: n=0

b0=1/(0+1)=1✓

Assume

bk = 1/(k+1)

Show

bk+1 = 1/(k+2)

Prove:

1/(k+2) ?= bk+1-1 / (1+bk+1-1)

=bk/(1+bk)

=(1/(k+1))/ (1+(1/(k+1)))

=1/[(1+k) (1+(1/(1+k)))]

=1/(k+2)

YAY

1/(k+2) = 1/(k+2) ✓

13) T(1)=a

T(n) = T(n/2)+b

T(n/2)=T(n/4)+b

* T(n)= T(n/4)+b +b = T(n/4) + 2b << first unrolling

T(n/4) = T(n/8)+b

* T(n) = T(n/8)+b +2b = T(n/8)+ 3b << second unrolling

T(n/8) = T((n/8)/2) +b = T(n/16) +b

* T(n) = T(n/16) +b +3b = T(n/16) + 4b << third unrolling

T(n)=T(n/2k+1)+(k+1)b

If we assume n/2k+1 =1

n=2k+1

Log n =k+1

1-logn =k

T(n)=T(1)+(k+1)b

T(n)= a +(k+1)b

T(n) = a +(1-log(n)+1)b

T(n)=a + blog(n)

O(logn)

14)

T(1)=a

T(n) = 2T(n/2)+b

T(n/2)=2T(n/4)+b

* T(n)=2(2T(n/4)+b)+b = 4T(n/4)+ 2b +b = 4T(n/4)+ 3b << first unrolling

T(n/4) = 2T(n/8)+b

* T(n)=4(2T(n/8)+b) +3b = 8T(n/8)+ 4b +3b = 8T(n/8)+ 7b << second unrolling

T(n)= 2k+1 T(n/2k+1)+ [(2k+1)-1]b

If we assume n/2k+1 =1

n=2k+1

T(n)= 2k+1 a+[(2k+1)-1]b

T(n)= n\* a+[n-1]b

T(n) = a\*n + b\*n –b

O(n)

15)

T(1) = a

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

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

* T(n) = T(n-2) +(n-1)k +nk << first unrolling

T(n-2)= T(n-3) + (n-2)k

* T(n)= T(n-3) + (n-2)k +(n-1)k +nk << second unrolling

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

* T (n)= T(n-4) +(n-3)k + (n-2)k +(n-1)k +nk  << Third unrolling

T(n)=T(n –(k+1)) + (n-k)k +(n(k-1))k + (n-(k-2))k + nk