CSCE 222 (Carlisle), Discrete Structures for Computing
Spring 2020
Homework 6

Type your name below the pledge to sign
On my honor, as an Aggie, I have neither given nor received unauthorized aid on this academic work.
ARTHUR CHEN

Instructions:

- The exercises are from the textbook. You are encouraged to work extra problems to aid in your learning; remember, the solutions to the odd-numbered problems are in the back of the book.
- Grading will be based on correctness, clarity, and whether your solution is of the appropriate length.
- Always justify your answers.
- Don't forget to acknowledge all sources of assistance in the section below, and write up your solutions on your own.
- *Turn in .pdf file to Gradescope by the start of class on Tuesday, February 25, 2020.* It is simpler to put each problem on its own page using the LaTeX clearpage command.

Help Received:

- List any help received here, or "NONE".
- NONE

Exercises for Section 3.3:

2: (1 point).
for a nested loop like this, the number of iteration is $n * n$.
for each of the iteration, addition is preformed twice.
So the total number of operation is $2 * n^2$
Which is $O(n^2)$

4: (1 point).
in this while loop, there are two operation for each iteration.
because of $i = 2 * i$ in each of the iteration, i grows by 1,2,4,8....
since the while loop stops when i reaches n, lets say there is a x makes $2^x < n$
since there won't be operation for the last x, x is the floor function of the logarithm with base of n.
$x = \lfloor log_2(n) \rfloor$
since i starts from 1, the number of i will be $\lfloor log_2(n) \rfloor + 1$
and the number of operation will be $2(\lfloor log_2(n) \rfloor) + 2$, thus O(log n)

8: (2 points).
to find x^{2^k}, we need square k times
$x, x^{2^1}, x^{2^2}, x^{2^3} ... x^{2^k}$
the number of operation is k

To find the same thing by multiplying x by itself, we need time x by $2^k - 1$ times.
the number of operation is $2^k - 1$

since $k < 2^k - 1$, the first method is more efficient. squaring x successively is more efficient way than multiplying x by itself.

10b: (2 points).
since this while loop is rewriting each of the 1's to 0's to achieve counting, the number of iteration is equal to the number of 1's that bit string has

12b: (2 points).
part a:
note that there is only one operation in each iteration, so the number of both are the same
In this triple nested loop, since the first layer operates n times, second layer operates $n-(i+1)+1 = n - i$, and third layer operates $j - (i + 1)$ times.
Since j is the number of operation of the second layer, so the total number of operation is $n * ((n - i)(n + i + 1)/2) - (i + 1)$, which is $O(n^3)$.

part b:

since the first loop executed at least from i to n/4 times, from 3n/4 to n times for the second loop, and (3n/4)-(n/4) = n/2 times for the third loop, the number of iteration is $(n/4)(n/4)(n/2) = n^3/32$.
so the complexity of the algorithm is $\Omega(n^3)$
from part a, the complexity is also proven to be $O(n^3)$
so the complexity is $\Theta(n^3)$

14a: (2 points)
through the given notation:
$c = 2; a_0 = 1; a_1 = 1; a_2 = 3; n = 2$
we set y as:
$y = a_n = a_2 = 3$
first iteration:
i=1; and y = 3*2+1=7
second iteration:
i=2; and y = 7*2+1=15
the this algorithm will output 15

14b: (2 points)
this algorithm is composed of one multiplication and one addition for every iteration in this for loop.
since the range of i is from 1 to n, there are n iterations.
therefore, the number of multiplication is 1*n = n, the number of addition is 1*n = n.

16(a-f): (4 points)
Each of the operation takes 10^{-11} seconds, and one day has $24 * 60 * 60 = 86400$ seconds.
So $86400/(10^{-11}) = 8.64 * 10^{15}$ operations can be preformed in one day.
a.
$logn = 8.64 * 10^{15}$, note that bits only have 2 possible values, so the logarithm is base of 2.
Then $n = 2^{8.64*10^{15}}$

b.
$1000 * n = 8.64 * 10^{15}$
Then $n = 8.64 * 10^{12}$

c.
$n^2 = 8.64 * 10^{15}$
Then $n = 9.295 * 10^7$

d.

$1000 * n^2 = 8.64 * 10^{15}$
Then $n = 2.93939 * 10^6$

e.
$n^3 = 8.64 * 10^{15}$
Then $n = 205197$

f.
$2^n = 8.64 * 10^{15}$
Then $n = 52$

20(b,c,e,g): (2 points)

b.
$log(2n) - log(n)$
$= log(2) + log(n) - log(n)$
$= log(2)$

c.
$100(2n)/(100n)$
$= 2$

e.
$((2n)^2)/(n^2)$
$= 4$

g.
$(2^{2n})/(2^n)$
$= (2^n * 2^n)/(2^n)$
$= 2^n$

42: (2 points)

procedure of this greedy algorithm:
assume s_1 is the start time of talks and e_1 is the end time of talks.
sort talks by finishing time and reorder so that $e_1 \leq e_2 \leq e_3 ... e_n$
$S := \emptyset$
for j=1 to n
if talk j is compatible with S then $S := S \cup \{talkj\}$
return S{S is the set of talks scheduled}
the sorting step at the start of the procedure needs $O(nlogn)$, which arrange the talks in increasing order of their talk time.

for the worst case, the loop will have to run n times to go through every case, which is O(n), but less than the one above.
so the complexity of the algorithm is O(n log n)