Name: ID#: X500: @umn.edu **A**

CS 2011: Practice Quiz 3 SOLUTION

Summer 2018

University of Minnesota

Quiz period: 15 minutes

Points available: 40

Use the following table of letter/number correspondences for encryption problems.

0	1	2	3	4	5	6	7	8	9	10	11	12
A	B	C	D	E	F	G	H	I	J	K	L	M
13	14	15	16	17	18	19	20	21	22	23	24	25
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Problem 1 (10 pts): Encrypt the following message using the Caesar cipher with the key provided.

- Message: `ZEBRA`
- Key: 8
- **Show some work or full credit.**
- Give the letters of the final encrypted message.

SOLUTION:

- $Z \equiv 25$, $25 + 8 \mod 26 = 33 \mod 26 = 7 \equiv H$
- $E \equiv 4$, $4 + 8 \mod 26 = 12 \mod 26 = 12 \equiv M$
- $B \equiv 1$, $1 + 8 \mod 26 = 9 \mod 26 = 9 \equiv J$
- $R \equiv 17$, $17 + 8 \mod 26 = 25 \mod 26 = 25 \equiv Z$
- $A \equiv 0$, $0 + 8 \mod 26 = 8 \mod 26 = 8 \equiv I$

Encrypted message: HMJZI

Problem 2 (10 pts): Arrange the following functions in order of big-O growth rate right with slowest growing functions listed first.

(A) $2n^3 \log n$
(B) $5n(\log n)^2$
(C) $4n + n!$
(D) $7\sqrt{n} + \log n$
(E) $3^n + n \log n$

SOLUTION:

- *(D)* $7\sqrt{n} + \log n$
- *(B)* $5n(\log n)^2$
- *(A)* $2n^3 \log n$
- *(E)* $3^n + n \log n$
- *(C)* $4n + n!$

Problem 3 (10 pts): Use a proof by induction to show that for integer $n > 4$, $2^n > n^2$. Clearly illustrate Base and Inductive Cases and identify the Inductive Hypothesis.
SOLUTION:

BASE CASE: For $n = 5$, $2^5 = 32$ *and* $5^2 = 25$. *Since* $32 > 25$ *the property holds.*

INDUCTION CASE:

1. *Assume that for* k, *the inequality* $2^k > k^2$ *holds.*
2. *Show for* $k + 1$, *the inequality* $2^{k+1} > (k + 1)^2$ *holds.*
3. *Starting with the left side of the inequality transform as follows.*

$$
\begin{aligned}
2^{k+1} &= 2 \cdot 2^k && \textit{Def of Exponentiation} \\
&> 2 \cdot k^2 && \textit{By Inductive hypothesis (1)} \\
&> k^2 + 4k && \textit{Because } k \geq 5 \\
&> k^2 + 2k + 1 && \textit{Because } k \geq 5 \\
&= (k + 1)^2 && \textit{Factor polynomial}
\end{aligned}
$$

4. *Have shown that left hand side of inequality is indeed larger than right hand side so induction case holds.*

Problem 4 (10 pts): The algorithm `complete(A)` determines if array `A` of length N contains exactly the numbers $0, 1, 2, \ldots, N - 1$ in some order in which case `true` is returned. Otherwise, if some number is missing, `false` is returned.
Analyze the code below and give a big-O estimate of its worst-case runtime.

```
 1: complete(A[] : int array){
 2:   for(n=0; n<length(A); n++){
 3:     found = false
 4:     i = 0
 5:     while(i < length(A)
 6:           and found == false)
 7:     {
 8:       if(A[i] == n){
 9:         found = true
10:       }
11:       i++;
12:     }
13:     if(found == false){
14:       return false;
15:     }
16:   }
17:   return true;
18: }
```

SOLUTION: The outer loop from line 2 to 16 iterates N times where N is the length of the array. The operations at lines 3 and 4 are constant time. The inner loop from lines 5 to 12 also runs for N iterations and each operation within the loop is constant time. These leads to the combined complexity of the loops being $O(N^2)$. *In the worst case the algorithm will check for all values 0,1,...,(N-1). This means the overall runtime is* $O(N^2)$.