

Latex Example

VE281 - Data Structures and Algorithms, Xiaofeng Gao, Autumn 2019

* Please upload your assignment to website. Contact webmaster for any questions.

* Name:_____ Student ID:_____ Email: _____

Please see the following samples for Latex applications.

1. We use “enumerate” to list questions. E.g., see this question: use minimal counterexample principle to prove that for every integer $n > 7$, there exist integers $i_n \geq 0$ and $j_n \geq 0$, such that $n = i_n \times 3 + j_n \times 5$.

Proof. We use “proof” environment to answer questions asking for PROOF. □

2. Show that the equation $f(m, n) = 2^m(2n + 1) - 1$ defines a one-to-one correspondence between $\omega \times \omega$ and ω .

Solution. We use “solution” environment to answer other questions. □

3. Check how to write in Latex, like:

- Symbols, like $a, b, c, \alpha, \beta, \gamma, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbb{R}, \mathbb{S}, \mathbb{T}, \mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{Z}$.
- Functions, like $\sin \theta, \max x, \lg n, \arg \max_i \exp i$.
- Formulae, like *in-line style* formula $a^2 + b^2 = c^2$, and *display style* formula

$$\sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}.$$

- Environments, like “enumerate”, “itemize”, “definition”, etc.

4. Learn Tables and Figures, E.g., fill in the blanks with either true or false:

$f(n)$	$g(n)$	$f = O(g)$	$f = \Omega(g)$	$f = \Theta(g)$
$100n^3 + 3n$	$100n^2 + 2n + 100$			
$50n + \log n$	$10n + \log \log n$			
$50n \log^2 n$	$n \log \log n$			
n^5	3^n			
$n!$	5^n			

5. Learn how to use **algorithm2e** package, like Alg. 1.

Algorithm 1: BUBBLESORT

input : An array $A[1 \dots n]$ of n elements.

output: $A[1 \dots n]$ in nondecreasing order.

```
1  $i \leftarrow 1$ ;  $sorted \leftarrow false$ ;
2 while  $i \leq n - 1$  and not  $sorted$  do
3    $sorted \leftarrow true$ ;
4   for  $j \leftarrow n$  downto  $i + 1$  do
5     if  $A[j] < A[j - 1]$  then
6       interchange  $A[j]$  and  $A[j - 1]$ ;
7        $sorted \leftarrow false$ ;
8    $i \leftarrow i + 1$ ;
```

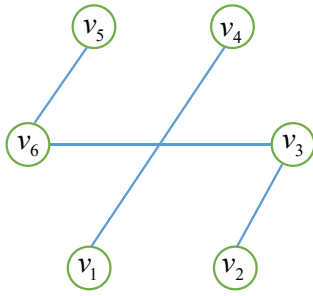


Figure 1: Graph G_1

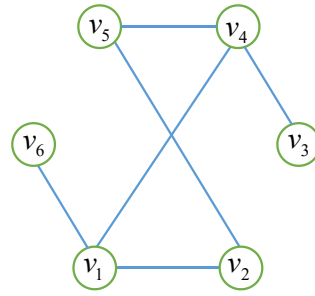


Figure 2: Graph G_2

- (a) What is the minimum number of element comparisons? When is this minimum achieved?

Solution. If having multiple sub-questions, put “solution” environment inside each sub-question. \square

- (b) What is the maximum number of element comparisons? When is this maximum achieved?
(c) Express the running time of Alg. 1 in terms of the O and Ω notations.
(d) Can the running time of the algorithm be expressed in terms of the Θ notation? Explain.

More examples about algorithm2e (Left \rightarrow source code; Right \rightarrow display in PDF):

1. **If block:**

```
\begin{algorithm} [H]
\KwIn{ $x$ ,  $y$ }
\KwOut{ $sign$ }
\BlankLine
\caption{ $div(x,y)$ } \label{Alg-div}
\If{ $rm(x,y)=0$ }{
     $sign=1$ ;
}
\Else{
     $sign=0$ ;
}
\Return{ $sign$ };
\end{algorithm}
```

Algorithm 2: $div(x,y)$

Input: x, y

Output: $sign$

```
1 if  $rm(x,y) = 0$  then
2   |  $sign \leftarrow 1$ ;
3 else
4   |  $sign \leftarrow 0$ ;
5 return  $sign$ ;
```

2. **If-ElseIf-Else block:**

```
\begin{algorithm}[H]
\KwIn{$score$}
\KwOut{Letter Grade}
\BlankLine
\caption{LetterGrade($score$)}
\label{Alg-Score}
\uIf{$score \ge 90$}{
    \textbf{output} $A$;
}
\uElseIf{$80 \le score < 90$}{
    \textbf{output} $B$;
}
\Else{
    \textbf{output} $P$;
}
\end{algorithm}
```

3. **While block:**

```
\begin{algorithm}[H]
\KwIn{$x$, $y$}
\KwOut{$x$}
\BlankLine
\While{$x \ge y$}{
    $x-=y$;
}
\textbf{output} $x$;
\end{algorithm}
```

4. **For block:**

```
\begin{algorithm}[H]
\KwIn{$n \in \mathbb{N}$}
\KwOut{The sum from 1 to $n$}
\BlankLine
\caption{Sum($n$)} \label{Alg-Sum}
$sum=0$;
\For{$temp=0$ to $n$}{
    $sum=sum+temp$;
}
\textbf{output} $sum$;
\end{algorithm}
```

Algorithm 3: LetterGrade(*score*)

Input: *score*

Output: Letter Grade

```
1 if score  $\geq 90$  then
2   |   output A;
3 else if  $80 \leq score < 90$  then
4   |   output B;
5 else
6   |   output P;
```

Algorithm 4: *rm*(*x*, *y*)

Input: *x*, *y*

Output: *x*

```
1 while  $x \geq y$  do
2   |    $x- = y$ ;
3 output x;
```

Algorithm 5: Sum(*n*)

Input: $n \in \mathbb{N}$

Output: The sum from 1 to *n*

```
1  $sum \leftarrow 0$ ;
2 for  $temp = 0$  to n do
3   |    $sum \leftarrow sum + temp$ ;
4 output sum;
```

5. **Repeat-Until block:**

```
\begin{algorithm}[H]
\KwIn{$a, b \in \mathbb{N}$}
\KwOut{Greatest common divide of $a$, $b$}
\BlankLine
\caption{GCD($a$, $b$)} \label{Alg-GCD}

\Repeat{$gcd=0$}{
    $gcd = a \bmod b$;
    $a=b$;
    $b=gcd$;
}
\textbf{output} $gcd$;
\end{algorithm}
```

6. **Case block:**

```
\begin{algorithm}[H]
\KwIn{$person$}
\KwOut{$person$'s gender}
\BlankLine
\caption{Gender} \label{Alg-Gender}
\Switch{$person$}{
\uCase{$person.gender=male$}{
\textbf{output} Male;
}
\uCase{$person.gender=female$}{
\textbf{output} Female;
}
\Other{
\textbf{output} Unknown;
}
}
\end{algorithm}
```

Algorithm 6: GCD(a, b)

Input: $a, b \in \mathbb{N}$

Output: Greatest common divisor of a, b

```
1 repeat
2   |  $gcd \leftarrow a \bmod b$ ;
3   |  $a \leftarrow b$ ;
4   |  $b \leftarrow gcd$ ;
5 until  $gcd = 0$ ;
6 output  $gcd$ ;
```

Algorithm 7: Gender

Input: $person$

Output: $person$'s gender

```
1 switch  $person$  do
2   | case  $person.gender = male$ 
3   |   do
4   |     output Male;
5   | case
6   |    $person.gender = female$ 
7   |     do
8   |       output Female;
9   | otherwise do
10  |   output Unknown;
```
