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. $\square$

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. $\square$

3. Check how to write in Latex, like:

   - Symbols, like a, b, c, α, β, γ, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbb{R}$, $\mathbb{S}$, $\mathbb{T}$, $\mathcal{U}$, $\mathcal{V}$, $\mathcal{W}$, $\mathscr{X}$, $\mathscr{Y}$, $\mathscr{Z}$.
   - 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.

```
i ← 1; sorted ← false;
while i ≤ n − 1 and not sorted do
    sorted ← true;
    for j ← n downto i + 1 do
        if A[j] < A[j − 1] then
            interchange A[j] and A[j − 1];
            sorted ← false;
    i ← i + 1;
```

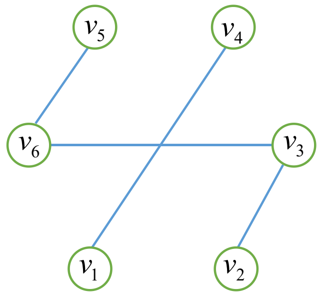

Figure 1: Graph G_1

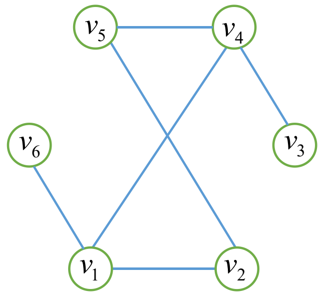

Figure 2: Graph G_2

(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. □

(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: $sign$

if $rm(x, y) = 0$ **then**
    $sign \leftarrow 1$;
else
    $sign \leftarrow 0$;
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}
```

Algorithm 3: LetterGrade($score$)

Input: $score$
Output: Letter Grade

if $score \geq 90$ **then**
    output A;
else if $80 \leq score < 90$ **then**
    output B;
else
    output P;

3. **While** block:

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

Algorithm 4: $rm(x, y)$

Input: x, y
Output: x

while $x \geq y$ **do**
    $x- = y$;
output x;

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 5: Sum(n)

Input: $n \in \mathbb{N}$
Output: The sum from 1 to n

$sum \leftarrow 0$;
for $temp = 0$ *to* n **do**
    $sum \leftarrow sum + temp$;
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{$gcd=0$}{
    $gcd = a \mod b$\;
    $a=b$\;
    $b=gcd$\;
}
\textbf{output} $gcd$\;
\end{algorithm}
```

Algorithm 6: GCD(a, b)

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

Output: Greatest common divisor of a, b

```
repeat
    gcd ← a mod b;
    a ← b;
    b ← gcd;
until gcd = 0;
output gcd;
```

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 7: Gender

Input: $person$

Output: $person$'s gender

```
switch person do
    case person.gender = male
     do
        output Male;
    case
     person.gender = female
     do
        output Female;
    otherwise do
        output Unknown;
```