算设

Hw 1 Feb 24, Spring 2022

gjm, 2020019999 gjm20@mails.tsinghua.edu.cn

1 标题左对齐

第一个 section 使用 `\firstsection`，避免切换到新的一页；后续使用 `\section`，会自动从新的一页开始 section。

1.1 子标题也左对齐

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

1.1.1 子子标题也左对齐

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

1.2 展示带圈字符

① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩ ⑪ ⑫ ⑬ ⑭ ⑮ ⑯ ⑰ ⑱ ⑲ ⑳ , ㊎

2 问题列表

整个文档使用同一个问题计数器，当然，也可以使用 \setcounte{ProblemCounter}{1} 重设计数。

2.1 双列问题列表

Problem 1

Prove: $2n + \Theta\left(n^2\right) = \Theta\left(n^2\right)$

Solution:

$\because$ 根据 Θ 定义

$\therefore \exists n_0, c_1, c_2,\ s.t.\ \forall n > n_0,\ c_1 n^2 \leqslant \Theta(n^2) \leqslant c_2 n^2$

$$\therefore c_1 + 2/n \leqslant \frac{2n + \Theta(n^2)}{n^2} \leqslant c_2 + 2/n$$

$$\therefore c_1 \leqslant \frac{2n + \Theta(n^2)}{n^2} \leqslant c_2 + 3$$

$$\therefore c_1 n^2 \leqslant 2n + \Theta(n^2) \leqslant (c_2 + 3) n^2$$

$$\therefore 2n + \Theta(n^2) = \Theta(n^2)$$

Problem 2

Prove: $\Theta(g(n)) \cap o(g(n)) = \varnothing$

Solution:

$\forall f \in \Theta(g),\ \exists c_2, n_0,\ s.t.\ \forall n > n_0,\ g \leqslant c_2 f$

若 $f \in o(g)$, 则 $\forall c,\ \exists n_1,\ s.t.\ \forall n > n_1,\ cf < g$

这与$f \in \Theta(g)$矛盾

$\therefore \Theta(g(n)) \cap o(g(n)) = \varnothing$

在这里使用 *\columnbreak* 强制换栏

Problem 3

Prove: $\Theta(g(n)) \cup o(g(n)) \neq O(g(n))$

Ref to book.

Solution:

直观理解三个符号:

$\Theta(g) \iff f/g$ 有界, 且下确界>0

$o(g) \iff f/g \to 0$

$O(g) \iff f/g \leqslant const$

$\therefore$ 设 $g = n^2,\ f = (1 + (-1)^n) n^2$

此时, $f = O(g)$, 但是$f \neq \Theta(g)$且$f \neq o(g)$

Problem 4

Prove: $\max(f(n), g(n)) = \Theta(f(n) + g(n))$

Solution:

$\because f, g \geqslant 0$

$\therefore max(f,\ g) \leqslant f + g \leqslant 2max(f,\ g)$

$\therefore c_1 = 1,\ c_2 = 2$

$\therefore \max(f(n), g(n)) = \Theta(f(n) + g(n))$

2.2 单列问题列表

Problem 1 **Relative asymptotic growths** CLRS, P61, 3-2

Indicate, for each pair of expressions (A, B) in the table below, whether A is O, o, Ω, ω, or Θ of B. Assume that $k \geq 1, \epsilon > 0$, and $c > 1$ are constants. Your answer should be in the form of the table with "yes" or "no" written in each box.

Solution:

	A	B	O	o	Ω	ω	Θ
a.	$\lg^k n$	n^ϵ	Y	Y	N	N	N
b.	n^k	c^n	Y	Y	N	N	N
c.	$\sqrt{n}$	$n^{\sin n}$	N	N	N	N	N
d.	2^n	$2^{n/2}$	N	N	Y	Y	N
e.	$n^{\lg c}$	$c^{\lg n}$	Y	N	Y	N	Y
f.	$\lg(n!)$	$\lg(n^n)$	Y	N	Y	N	Y

Problem 2 **Ordering by asymptotic growth rates** CLRS, P61, 3-3

a. Rank the following functions by order of growth; that is, find an arrangement $g_1, g_2, \ldots, g_{30}$ of the functions satisfying $g_1 = \Omega(g_2), g_2 = \Omega(g_3), \ldots, g_{29} = \Omega(g_{30})$. Partition your list into equivalence classes such that functions $f(n)$ and $g(n)$ are in the same class if and only if $f(n) = \Theta(g(n))$.

$\lg(\lg^* n)$	$2^{\lg^* n}$	$(\sqrt{2})^{\lg n}$	n^2	$n!$	$(\lg n)!$
$(\frac{3}{2})^n$	n^3	$\lg^2 n$	$\lg(n!)$	2^{2^n}	$n^{1/\lg n}$
$\ln\ln n$	$\lg^* n$	$n \cdot 2^n$	$n^{\lg\lg n}$	$\ln n$	1
$2^{\lg n}$	$(\lg n)^{\lg n}$	e^n	$4^{\lg n}$	$(n+1)!$	$\sqrt{\lg n}$
$\lg^*(\lg n)$	$2^{\sqrt{2}\lg n}$	n	2^n	$n \lg n$	$2^{2^{n+1}}$

b. Give an example of a single nonnegative function $f(n)$ such that for all functions $g_i(n)$ in part (a), $f(n)$ is neither $O(g_i(n))$ nor $\Omega(g_i(n))$

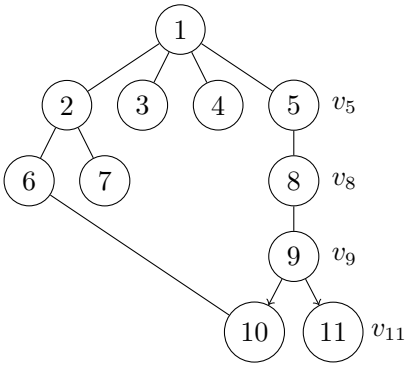
Solution:

a. $a > b$ 表示 $a = \Omega(b)$, $a = b$ 表示 $a = \Theta(b)$。顺序为: $2^{2^{n+1}} > 2^{2^n} > (n+1)! > n! > e^n > n \cdot 2^n > 2^n > (3/2)^n > (\lg n)^{\lg n} = n^{\lg\lg n} > (\lg n)! > n^3 > n^2 = 4^{\lg n} > n \ln n = \lg n! > n = 2^{\lg n} > (\sqrt{2})^{\lg n} > 2^{\sqrt{2\lg n}} > \lg^2 n > \ln n > \sqrt{\lg n} > \ln\ln n > 2^{\ln^* n} > \lg^* n = \lg^*(\lg n) > \lg(\lg^* n) > n^{1/\lg n} = 1$

b. $f(n) = (1 + (-1)^n)\, n$

3 绘制图、树

如果要绘制图、树，使用 LuaLaTeX 编译，否则使用 XeLaTeX 编译，要快一些。



更多例子可见 https://texample.net/

4 伪代码示意

Algorithm 1 Subproblem Optimize

```
if r − l ⩽ 30 then
    sort p.pointlist by a specific sort algorithm
    calculate minimal distance brutely here
    return minimal distance
else if l ⩾ r − 1 then
    return INF
```

引用代码使用 *Algorithm 1*。如果要允许代码块换行，使用 breakablealgorithm 代替 algorithm。

5 代码渲染

这里展示文档内代码渲染功能，支持内插代码和从文件读取代码。

Example Code 1

```cpp
// This is an example of code block.
// Language: C++
#include <iostream>
using namespace std;
int main ()
{
        cout << "Hello World!" << endl;
        return 0;
}
```

Load from file

```python
"""
一个无聊的生成四则运算表达式的小程序
"""
from random import random, randint

toStrFunc = 0

class Node:
    _randMax = 100
    _numtype = 'num'
    _nodetype = 'node'

    type = None
    num = 0
    lson = None
    rson = None
    op = ''

    def __init__(self) -> None:
        self.type = self._numtype
        self.num = int(random() * self._randMax)
        self.op = '+-*/'[randint(0, 3)]

    def __str__(self) -> str:
        if self.type == self._numtype:
            return str(self.num)
        elif self.type == self._nodetype:
            global toStrFunc
            if toStrFunc == 1:
                return f'({str(self.lson)}{self.op}{self.rson})'

            exp = ''
```

```
            if self.lson.priority() < self.priority(): exp = f'({self.lson}){self.op}'
            else: exp = f'{self.lson}{self.op}'
            if self.rson.priority() < self.priority(): exp = f'{exp}({self.rson})'
            else: exp = f'{exp}{self.rson}'
            return exp
        else:
            return 'NULL NODE!!!'

    def expand(self):
        if self.type == self._numtype:
            self.type = self._nodetype
            self.lson = Node()
            self.rson = Node()
        return self

    def priority(self):
        if self.type == self._numtype: return 1000
        else:
            if self.op in '+-': return 10
            elif self.op in '*/': return 20


def dfs(h: Node, k: int):
    while k > 0:
        k -= 1
        h.expand()
        h = [h.lson, h.rson][randint(0, 1)]


for i in range(10):
    h = Node()
    dfs(h, 5)
    print(h)
```