

Master Theorem.sage

February 22, 2022

1 Master Theorem Recurrences

1.1 Maxwell Kapral

<sub>

<i>powered by SageMath in Jupyter</i>

</sub>

1.2 Of the Form:

$$T(n) = aT\left(\frac{n}{b}\right) + \mathcal{O}(n^k \log^p(n))$$

Where n is the size of the problem, a is the number of subproblems in the recursion $a \geq 1$, $\frac{n}{b}$ is the size of each subproblem, $b > 1$, $k \geq 0$, $p \in \mathbb{R}$

expr is a lazy, magic variable. If expr is some $f(n)$, then $T(n) \in \mathcal{O}(f(n))$ **if** $f(n) \in \Omega(n^{\log_b(a)+c_1})$ **and** $a \cdot f(\frac{n}{b}) \leq c_2 \cdot f(n)$ **where** $c_1 > 0$ **and** $c_2 < 1$, for all n large enough $(+\infty)$.

```
[1]: from sage.rings.asymptotic.term_monoid import OTermMonoid, ExactTermMonoid
from sage.rings.asymptotic.term_monoid import DefaultTermMonoidFactory as T_M
from sage.rings.asymptotic.growth_group import GrowthGroup
n = SR.var('n')

def mastertheorem(a, b, k, p, expr=n):
    assert(a >= 1), "a not greater than or equal to 1"
    assert(b > 1), "b not greater than 1"
    assert(k >= 0), "k not greater than or equal to 0"
    assert(p in RR), "p must be a real number"
    if expr != n.factorial():
        G_G = GrowthGroup('(RR_+)^n * n^RR * log(n)^RR')
        bigO = OTermMonoid(T_M, growth_group=G_G, coefficient_ring=RR)
        bigTheta = ExactTermMonoid(T_M, growth_group=G_G, coefficient_ring=RR)
        if expr == n:
            if p != 0:
                expr = (n**k)*(log(n)**p)
            else:
                expr = n**k
```

```

ans = None

eps = 0.01
if bigO(n**(log(a, b) - eps)).can_absorb(bigO(expr)):
    ans = n**log(a, b)
if ans is None:
    if a >= b**k:
        if p > -1:
            for i in range(0, p+1, 1):
                if bigTheta((n**log(a, b))*(log(n)**i)).
↪can_absorb(bigTheta(expr)):
                    ans = (n**log(a, b))*(log(n)**(i+1))
                    break
            elif p == -1:
                ans = (n**log(a, b))*log(log(n), b)
            elif p < -1:
                ans = n**log(a, b)
if ans is None:
    c1 = 0.01
    if bigO(expr).can_absorb(bigO(n**(log(a, b) + c1))):
        c2 = 0.99
        if bigO(c2*expr).can_absorb(bigO(a*(expr/b))):
            ans = expr
else:
    ans = factorial(n)
#####
# OUTPUT #
#####
if a == 1:
    if p == 0:
        show(LatexExpr(r"T\left(n\right)=T\left(\dfrac{n}{b}\right)+
r"}\right)+\log{n}"))
    elif p == 1:
        if k > 0:
            show(LatexExpr(r"T\left(n\right)=T\left(\dfrac{n}{b}\right)+
r"}\right)+\log{n}"))
        else:
            show(LatexExpr(r"T\left(n\right)=T\left(\dfrac{n}{b}\right)+
r"}\right)+\log{n}"))
    elif p == (-1):
        show(LatexExpr(r"T\left(n\right)=T\left(\dfrac{n}{b}\right)+
r"}\right)+\log{n}"))
    elif p > 1:
        if k > 0:
            show(LatexExpr(r"T\left(n\right)=T\left(\dfrac{n}{b}\right)+
r"}\right)+\log{n}"))
        else:

```

```

        show(LatexExpr(r"T\left(n\right)=T\left(\dfrac{n}{r}\right)+\log^{p}(r^n)"))
    elif p < (-1):
        show(LatexExpr(r"T\left(n\right)=T\left(\dfrac{n}{r}\right)+\log^{p}(r^n)"))
    else:
        if p == 0:
            show(LatexExpr(r"T\left(n\right)="+latex(a)+
                r"\thinspace T\left(\dfrac{n}{r}\right)+\log^{p}(r^n)"))
        elif p == 1:
            if k > 0:
                show(LatexExpr(r"T\left(n\right)="+latex(a)+
                    r"\thinspace T\left(\dfrac{n}{r}\right)+\log^{p}(r^n)"))
            else:
                show(LatexExpr(r"T\left(n\right)="+latex(a)+
                    r"\thinspace T\left(\dfrac{n}{r}\right)+\log^{p}(r^n)"))
        elif p == (-1):
            show(LatexExpr(r"T\left(n\right)="+latex(a)+
                r"\thinspace T\left(\dfrac{n}{r}\right)+\log^{p}(r^n)"))
        elif p > 1:
            if k > 0:
                show(LatexExpr(r"T\left(n\right)="+latex(a)+
                    r"\thinspace T\left(\dfrac{n}{r}\right)+\log^{p}(r^n)"))
            else:
                show(LatexExpr(r"T\left(n\right)="+latex(a)+
                    r"\thinspace T\left(\dfrac{n}{r}\right)+\log^{p}(r^n)"))
        elif p < (-1):
            show(LatexExpr(r"T\left(n\right)="+latex(a)+
                r"\thinspace T\left(\dfrac{n}{r}\right)+\log^{p}(r^n)"))
    print('') # Newline
    show(LatexExpr(r"T\left(n\right)\in\Theta\left("+latex(ans.simplify())+
        r"\right)"))

```

2 Examples

2.0.1 The cost $T(n)$ of a merge sort on a list of n numbers is governed by the following recurrence relation:

$$T(n) = 2T\left(\frac{n}{2}\right) + n - 1, \quad T(1) = 0$$

[2]:

```
# a = 2, b = 2, k = 1, p = 0
mastertheorem(2, 2, 1, 0)
```

$$T(n) = 2T\left(\frac{n}{2}\right) + n$$

$$T(n) \in \Theta(n \log(n))$$

2.0.2 Prove that

$$T(n) \in \mathcal{O}(n^2)$$

2.0.3 For

$$T(n) = 3T\left(\frac{n}{5}\right) + n^2 \quad n > 5$$

$$T(5) = 51 \quad n = 5$$

[3]:

```
# a = 3, b = 5, k = 2, p = 0
mastertheorem(3, 5, 2, 0)
```

$$T(n) = 3T\left(\frac{n}{5}\right) + n^2$$

$$T(n) \in \Theta(n^2)$$

2.0.4 Express the Running Time of the Following Recurrence Relation

$$T(n) = 5T\left(\frac{n}{3}\right) + n^4 \quad n > 1$$

$$T(1) = 1 \quad n = 1$$

[4]:

```
# a = 5, b = 3, k = 4, p = 0
mastertheorem(5, 3, 4, 0)
```

$$T(n) = 5T\left(\frac{n}{3}\right) + n^4$$

$$T(n) \in \Theta(n^4)$$

[5]: `# a = 3, b = 2, k = 2, p = 0`
`mastertheorem(3, 2, 2, 0)`

$$T(n) = 3T\left(\frac{n}{2}\right) + n^2$$

$$T(n) \in \Theta(n^2)$$

[6]: `# a = 4, b = 2, k = 2, p = 0`
`mastertheorem(4, 2, 2, 0)`

$$T(n) = 4T\left(\frac{n}{2}\right) + n^2$$

$$T(n) \in \Theta(n^2 \log(n))$$

[7]: `# a = 1, b = 2, k = 0, p = 0, expr = 2^n`
`mastertheorem(1, 2, 1, 0, (2**n))`

$$T(n) = T\left(\frac{n}{2}\right) + 2^n$$

$$T(n) \in \Theta(2^n)$$

[8]: `# a = 2, b = 2, k = 1, p = -1`
`mastertheorem(2, 2, 1, -1)`

$$T(n) = 2T\left(\frac{n}{2}\right) + \frac{n}{\log(n)}$$

$$T(n) \in \Theta\left(\frac{n \log(\log(n))}{\log(2)}\right)$$

[9]: `# a = 16, b = 4, k = 1, p = 0, expr = n!`
`mastertheorem(16, 4, 1, 0, factorial(n))`

$$T(n) = 16T\left(\frac{n}{4}\right) + n!$$

$$T(n) \in \Theta(n!)$$

[10]: `# a = 6, b = 3, k = 2, p = 1`
`mastertheorem(6, 3, 2, 1)`

$$T(n) = 6T\left(\frac{n}{3}\right) + n^2 \log(n)$$

$$T(n) \in \Theta(n^2 \log(n))$$

[11]: `# a = 4, b = 2, k = 0, p = 1`
`mastertheorem(4, 2, 0, 1)`

$$T(n) = 4T\left(\frac{n}{2}\right) + \log n$$

$$T(n) \in \Theta(n^2)$$

[12]: `# a = 2, b = 2, k = 1, p = 1`
`mastertheorem(2, 2, 1, 1)`

$$T(n) = 2T\left(\frac{n}{2}\right) + n \log(n)$$

$$T(n) \in \Theta(n \log(n)^2)$$

[13]: `# a = 2, b = 4, k = 0.51, p = 0`
`mastertheorem(2, 4, 0.51, 0)`

$$T(n) = 2T\left(\frac{n}{4}\right) + n^{0.51}$$

$$T(n) \in \Theta(n^{0.51})$$

[14]: `# a = sqrt(2), b = 2, k = 0, p = 1`
`mastertheorem(sqrt(2), 2, 0, 1)`

$$T(n) = \sqrt{2}T\left(\frac{n}{2}\right) + \log n$$

$$T(n) \in \Theta(\sqrt{n})$$

[15]: `# a = 3, b = 3, k = 1/2, p = 0`
`mastertheorem(3, 3, 1/2, 0)`

$$T(n) = 3T\left(\frac{n}{3}\right) + \sqrt{n}$$

$$T(n) \in \Theta(n)$$

[16]: `# a = 3, b = 2, k = 1, p = 0`
`mastertheorem(3, 2, 1, 0)`

$$T(n) = 3T\left(\frac{n}{2}\right) + n$$

$$T(n) \in \Theta\left(n^{\frac{\log(3)}{\log(2)}}\right)$$

[17]: `# a = 2, b = 7, k = 6, p = 0`
`mastertheorem(2, 7, 6, 0)`

$$T(n) = 2T\left(\frac{n}{7}\right) + n^6$$

$$T(n) \in \Theta(n^6)$$

[18]: `# a = 16, b = 4, k = 2, p = 3`
`mastertheorem(16, 4, 2, 3)`

$$T(n) = 16T\left(\frac{n}{4}\right) + n^2 \log(n)^3$$

$$T(n) \in \Theta(n^2 \log(n)^4)$$

[19]: `# a = 9, b = 2, k = 3, p = 0`
`mastertheorem(9, 2, 3, 0)`

$$T(n) = 9T\left(\frac{n}{2}\right) + n^3$$

$$T(n) \in \Theta\left(n^{\frac{2 \log(3)}{\log(2)}}\right)$$

[20]: `# a = 27, b = 3, k = 4, p = 0`
`mastertheorem(27, 3, 4, 0)`

$$T(n) = 27 T\left(\frac{n}{3}\right) + n^4$$

$$T(n) \in \Theta(n^4)$$

[21]: `# a = 16, b = 2, k = 4, p = 5`
`mastertheorem(16, 2, 4, 5)`

$$T(n) = 16 T\left(\frac{n}{2}\right) + n^4 \log(n)^5$$

$$T(n) \in \Theta(n^4 \log(n)^6)$$

[22]: `# a = 1, b = 2, k = 1, p = -1`
`mastertheorem(1, 2, 1, -1)`

$$T(n) = T\left(\frac{n}{2}\right) + \frac{n}{\log(n)}$$

$$T(n) \in \Theta\left(\frac{n}{\log(n)}\right)$$

[23]: `# a = 16, b = 2, k = 4, p = -1`
`mastertheorem(16, 2, 4, -1)`

$$T(n) = 16 T\left(\frac{n}{2}\right) + \frac{n^4}{\log(n)}$$

$$T(n) \in \Theta\left(\frac{n^4 \log(\log(n))}{\log(2)}\right)$$

[24]: `# a = 16, b = 2, k = 4, p = -4`
`mastertheorem(16, 2, 4, -4)`

$$T(n) = 16 T\left(\frac{n}{2}\right) + \frac{n^4}{\log(n)^4}$$

$$T(n) \in \Theta(n^4)$$

2.0.5 The Camerini Minimum Bottleneck Spanning Tree Algorithm

$$T(n) = T\left(\frac{n}{2}\right) + \mathcal{O}(n)$$

[25]: `# a = 1, b = 2, k = 1, p = 0`
`mastertheorem(1, 2, 1, 0)`

$$T(n) = T\left(\frac{n}{2}\right) + n$$

$$T(n) \in \Theta(n)$$

□