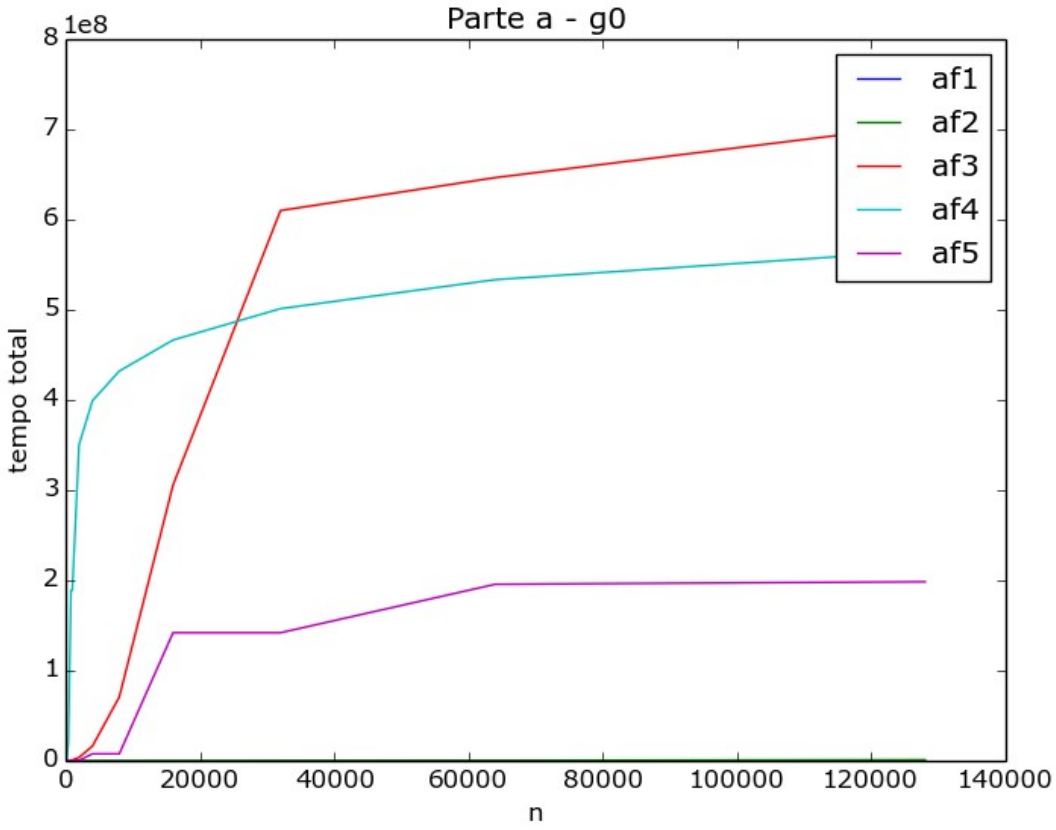
Trabalho 4

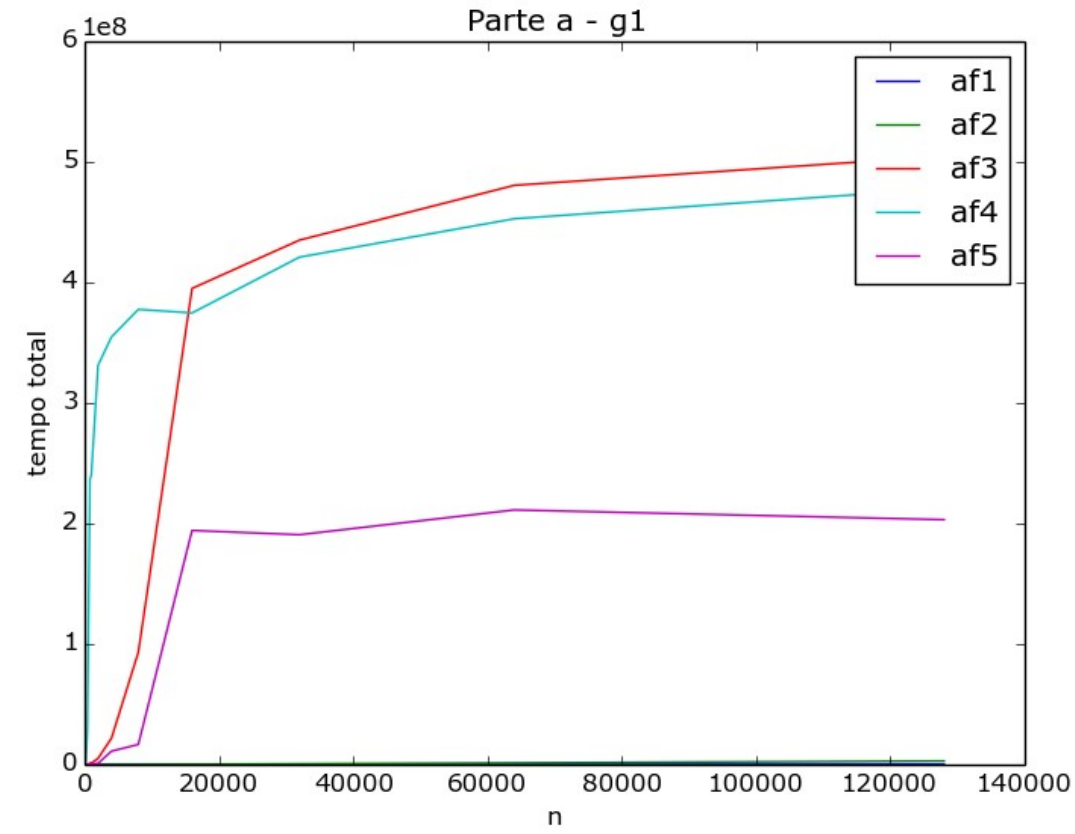
Universidade de Brasília
Departamento de Ciência da Computação
Disciplina: Projeto e Análise de Algoritmos
Código da Disciplina: 117536

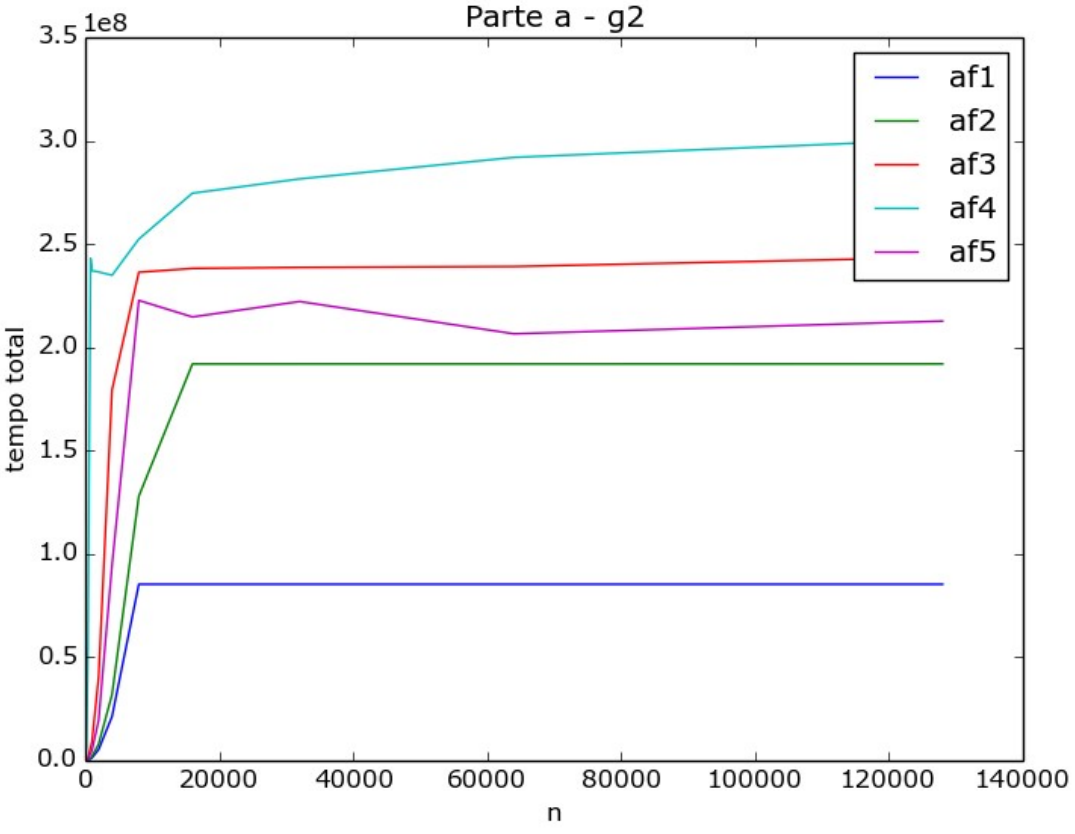
Aluno: Gabriel Martins de Miranda 13/0111350

1. Gráficos (todos os gráficos em graph/)

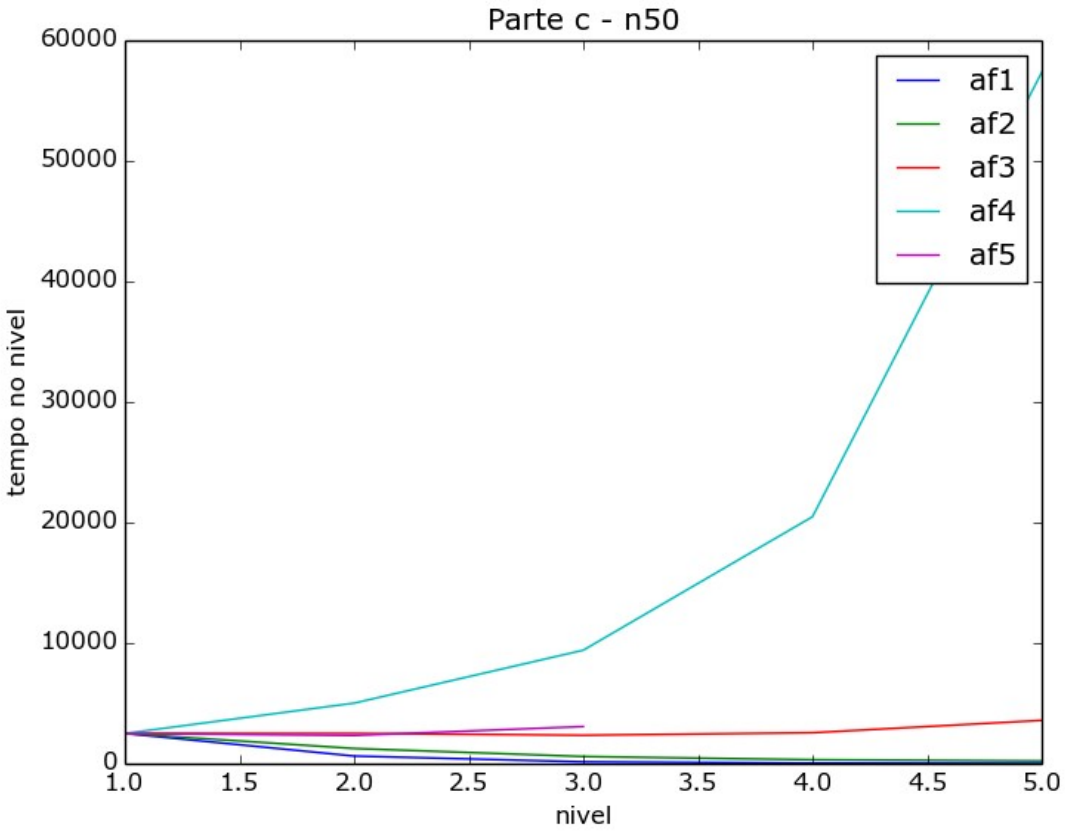
Etapa a

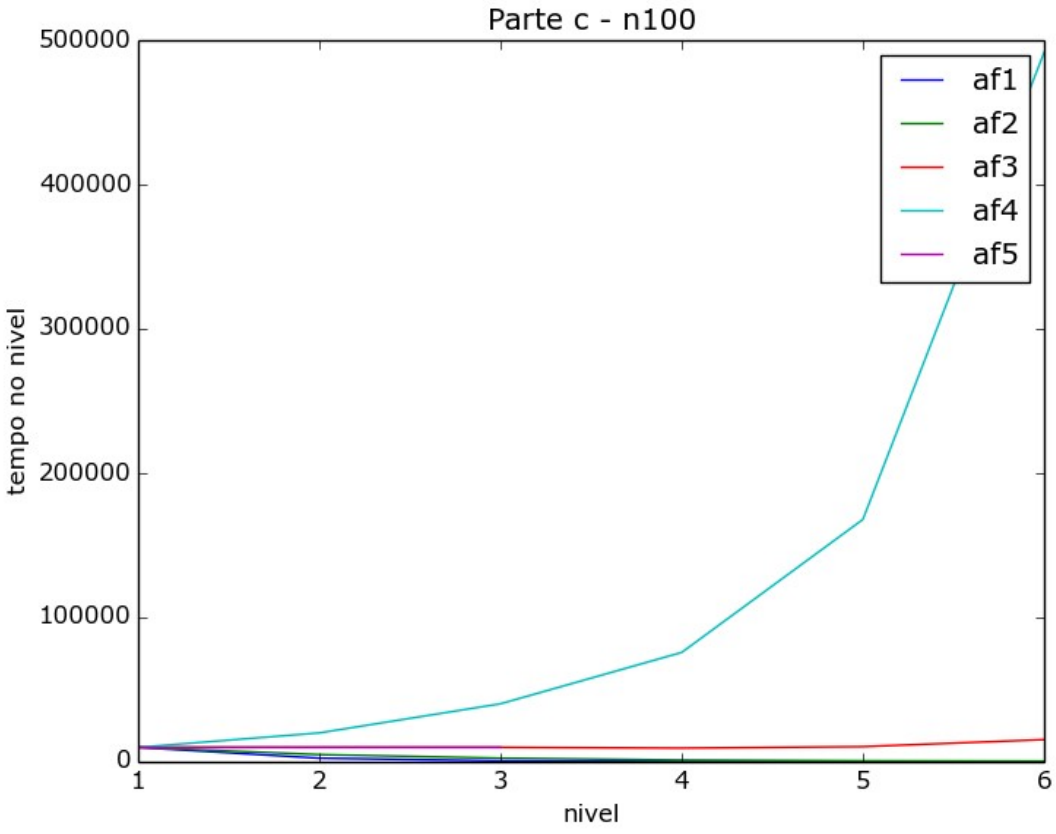


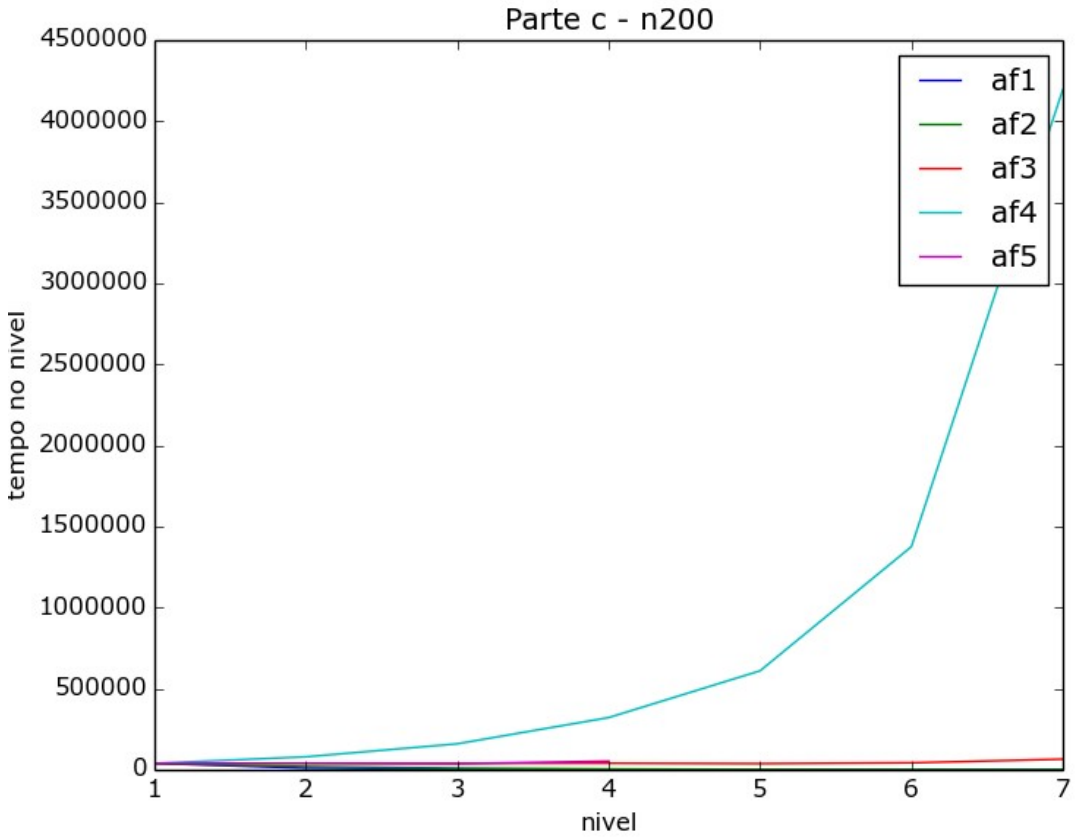

Parte a - g1
1e8
6
5
4
3
2
1
0
tempo total
0
20000
40000
60000
80000
100000
120000
140000
n
af1
af2
af3
af4
af5


Parte a - g2
1e8
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
tempo total
0
20000
40000
60000
80000
100000
120000
140000
n
af1
af2
af3
af4
af5

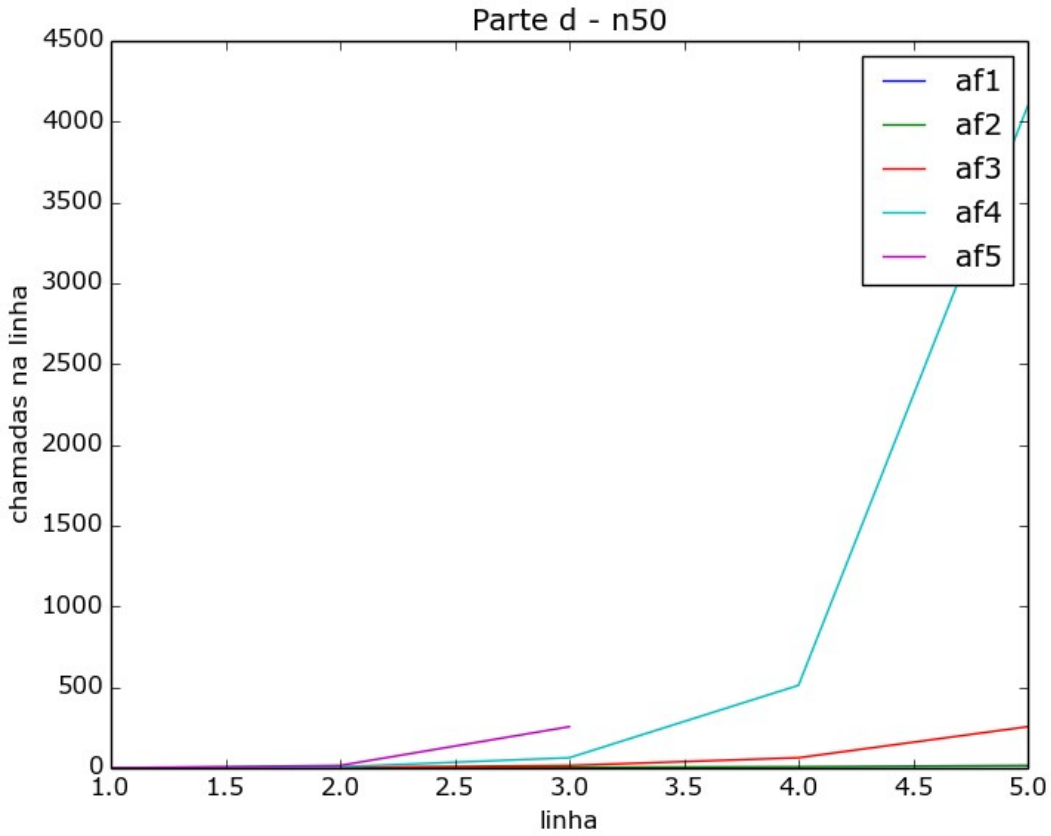
Etapa c

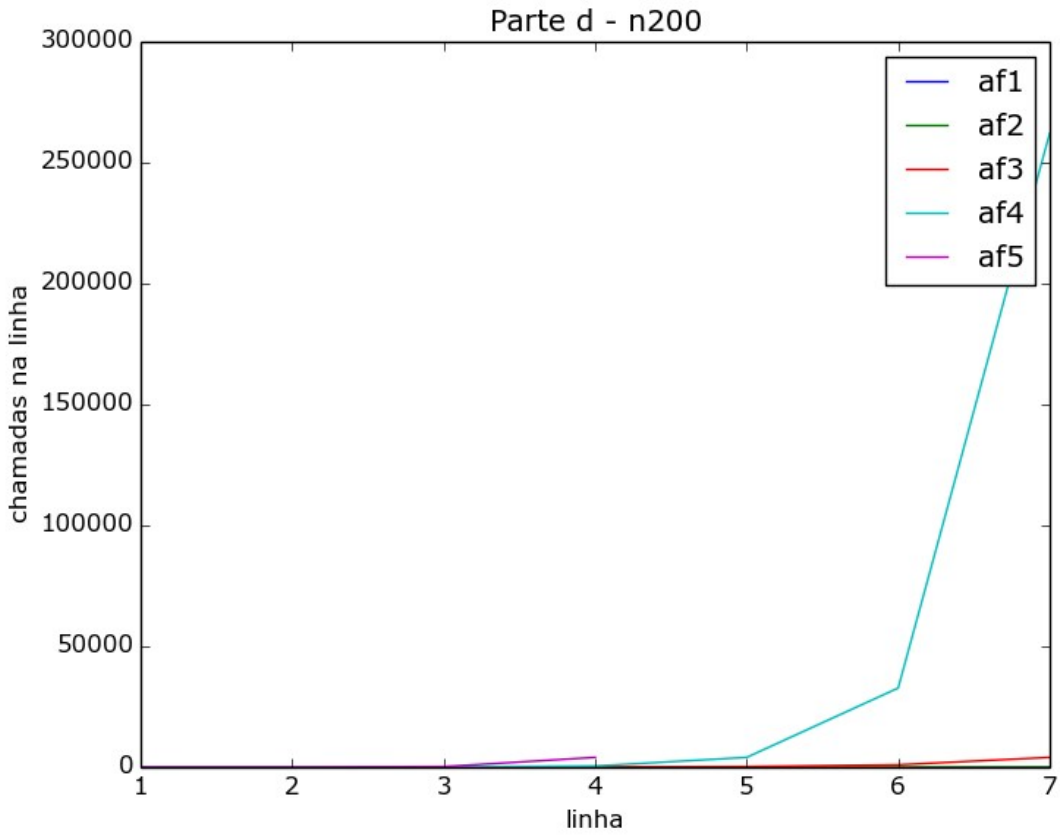

Parte c - n50
af1
af2
af3
af4
af5
tempo no nivel
nivel
0
10000
20000
30000
40000
50000
60000
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0


Parte c - n100
af1
af2
af3
af4
af5
tempo no nivel
nivel
0
100000
200000
300000
400000
500000
1
2
3
4
5
6


Parte c - n200
4500000
4000000
3500000
3000000
2500000
2000000
1500000
1000000
500000
0
tempo no nivel
1
2
3
4
5
6
7
nivel
af1
af2
af3
af4
af5

Etapa d


Parte d - n50
4500
4000
3500
3000
2500
2000
1500
1000
500
0
chamadas na linha
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
linha
af1
af2
af3
af4
af5



2. Respostas

1) Nos casos em que o método mestre se aplica:

Levando em consideração a forma T(n) = aT(n/b) + f(n)

Caso 1 – Tempo Esperado = n^1 = 50

T(n) = 2T((n/2) -0) + 0T((n/1) -0) + n^0
Com n = 50
|=> Altura arvore = 5
|=> Trabalho por nivel
Nivel 1 = 2
Nivel 2 = 6
Nivel 3 = 16
Nivel 4 = 40
Nivel 5 = 96
|=> Numero recursoes por nivel
Nivel 1 = 1
Nivel 2 = 2
Nivel 3 = 4
Nivel 4 = 8
Nivel 5 = 16
|=> Tempo total = 160
|=> Trabalho de fn por nivel
Nivel 1 = 1
Nivel 2 = 2
Nivel 3 = 4
Nivel 4 = 8
Nivel 5 = 16
|=> Metodo Mestre
T(n) = 2T((n/2) + n^0

b = 2, a = 2, f(n) = n^0
log de 2 na base 2 = 1
Caso 1 Método Mestre
T(n) = Teta(n^1)
|=>Tempo Execução = 2.4e-05s (normal)

Caso 2 – Tempo Esperado = lgn = 6.7

T(n) = 1T((n/2) -0) + 0T((n/1) -0) + n^0
Com n = 100
|=> Altura arvore = 6
|=> Trabalho por nivel
Nivel 1 = 2
Nivel 2 = 3
Nivel 3 = 4
Nivel 4 = 5
Nivel 5 = 6
Nivel 6 = 7
|=> Numero recursoes por nivel
Nivel 1 = 1
Nivel 2 = 1
Nivel 3 = 1
Nivel 4 = 1
Nivel 5 = 1
Nivel 6 = 1
|=> Tempo total = 27
|=> Trabalho de fn por nivel
Nivel 1 = 1
Nivel 2 = 1
Nivel 3 = 1
Nivel 4 = 1
Nivel 5 = 1
Nivel 6 = 1
|=> Metodo Mestre
T(n) = 1T((n/2) + n^0
b = 2, a = 1, f(n) = n^0
log de 1 na base 2 = 0
Caso 2 Método Mestre
T(n) = Teta(n^0lgn)
|=>Tempo Execução = 3.1e-05s (normal)

Caso 3 – Tempo Esperado = n = 50

T(n) = 1T((n/2) -0) + 0T((n/1) -0) + n^1
Com n = 50
|=> Altura arvore = 5
|=> Trabalho por nivel
Nivel 1 = 51
Nivel 2 = 27
Nivel 3 = 15
Nivel 4 = 10
Nivel 5 = 8
|=> Numero recursoes por nivel
Nivel 1 = 1
Nivel 2 = 1
Nivel 3 = 1
Nivel 4 = 1
Nivel 5 = 1
|=> Tempo total = 111
|=> Trabalho de fn por nivel
Nivel 1 = 50
Nivel 2 = 25
Nivel 3 = 12
Nivel 4 = 6
Nivel 5 = 3
|=> Metodo Mestre
T(n) = 1T((n/2) + n^1
b = 2, a = 1, f(n) = n^1
log de 1 na base 2 = 0
Caso 3 Método Mestre
T(n) = Teta(n^1)
|=>Tempo Execução = 9.4e-05s (normal)

a) Como os tempos gastos se relacionam com o método mestre ?

Os tempos gastos por nível se relacionam de acordo com o caso do método mestre. Para as execuções de caso 1, o tempo menor se concentra na raiz e vai crescendo para as folhas. No caso 2, se mantém praticamente constante e no caso 3, o peso maior é na raiz e vai caindo para as folhas.

b) De acordo com o método mestre qual era o tempo esperado? O tempo obtido foi maior ou menor? A diferença pode ser explicada com uma multiplicação por constante?

Para o caso 1, o tempo esperado é praticamente o tempo das folhas. No caso 2, é praticamente a altura da árvore pelo trabalho médio entre os níveis. Já no 3, é praticamente o tempo da raiz. Pelos gráficos, o tempo obtido foi geralmente menor, mas com uma curva de aparência semelhante ao tempo real, logo explicada por uma multiplicação por constante.

c) Como o número de linhas na árvore de recursão e o número de chamadas a funções em cada linha está relacionado ao método mestre?

Como a árvore gerada pelo Método Mestre é sempre simétrica, o número de linhas na árvore está de acordo com o tamanho da entrada em cada recursão (quando b é 2, a altura é lg n). Já o número de chamadas na linha está de acordo com o número de nós na linha vezes o número de chamadas recursivas (a).

d) Como os tempos de execução mudam em cada uma das linhas da árvore para cada um dos casos?

No caso 1, os tempos crescem da raiz as folhas. No 2, se mantém quase constante e no 3 diminui da raiz às folhas.

2) Para os casos executados, calculando o tamanho total da árvore em termos de número de chamadas recursivas e tempo por cada chamada, o tempo total está de acordo com o esperado?

O tempo total pode ser dado pelo número de chamadas recursivas na árvore multiplicado pelo tempo de cada chamada.

3) Nos casos em que a árvore de recursão é desbalanceada, o que acontece com o tempo por nível? O que acontece com o tempo para cada f(n)? O tempo f(n) aumenta ou diminui quando n cresce?

Execução com árvore desbalanceada

T(n) = 1T((n/4) -0) + 1T((n/2) -0) + n^0
Com n = 64000
|=> Altura arvore = 15
|=> Trabalho por nivel
Nivel 1 = 2
Nivel 2 = 6
Nivel 3 = 16
Nivel 4 = 40
Nivel 5 = 96
Nivel 6 = 224
Nivel 7 = 512
Nivel 8 = 1152
Nivel 9 = 2470
Nivel 10 = 4202
Nivel 11 = 4632
Nivel 12 = 3016
Nivel 13 = 1106
Nivel 14 = 210
Nivel 15 = 16
|=> Numero recursoes por nivel
Nivel 1 = 1
Nivel 2 = 2
Nivel 3 = 4
Nivel 4 = 8
Nivel 5 = 16
Nivel 6 = 32
Nivel 7 = 64
Nivel 8 = 128
Nivel 9 = 247
Nivel 10 = 382
Nivel 11 = 386
Nivel 12 = 232
Nivel 13 = 79
Nivel 14 = 14
Nivel 15 = 1
|=> Tempo total = 17700
|=> Trabalho de fn por nivel
Nivel 1 = 1
Nivel 2 = 2
Nivel 3 = 4
Nivel 4 = 8
Nivel 5 = 16
Nivel 6 = 32
Nivel 7 = 64
Nivel 8 = 128
Nivel 9 = 247
Nivel 10 = 382
Nivel 11 = 386
Nivel 12 = 232
Nivel 13 = 79
Nivel 14 = 14
Nivel 15 = 1
|=>Tempo Execução = 0.000192s (normal)

O tempo por nível pode apresentar uma ruptura em seu padrão devido ao desbalanceamento da árvore. O tempo pode crescer e logo em seguida diminuir, assim como no caso acima. Em muitas casos o tempo aumentou.

4) Nas recursões em que os valores de c ou f são diferentes de zero, os resultados obtidos estão de acordo com as complexidades esperadas?

Os resultados estão de acordo quando calculados com os métodos iterativo ou substituição, já que o método mestre não pode ser usado. Nestes casos o tamanho da árvore explodiu e não foi possível representar em gráfico.

5) Para os casos onde não foi possível executar a recursão até o final, faça uma estimativa, baseada na teoria, de quanto tempo demoraria para rodar a recursão.

Muitas casos não puderam ser executados até o final. Este foi um grande problema do projeto. Mesmo setando um tempo máximo de execução, a recursão gasta um tempo maior que o esperado para fazer os retornos. Alguns casos demorariam mais de 3 horas de execução. A solução utilizada foi usar o sinal alarme e mostrar os dados obtidos até o momento. Pode-se usar os métodos aprendidos em aula, como iterativo.