Atividade Prática 10

Aluno: Raphael Henrique Braga Leivas

Matrícula: 2020028101

1 OBJETIVO

O objetivo da prática é visualizar os efeitos que diferentes parâmetros m e CARGA tem em uma árvore B simulada, em especial do tempo de execução.

2 METODOLOGIA

Temos 2 parâmetros para configurar:

- m: indica o mínimo de registros em um nó (m) e o mínimo de filhos do nó (m+1). Além disso, especifica o máximo registros (2m) e o máximo de filhos (2m+1) do nó;
- CARGA: indica o tamanho de cada registro salvo na árvore B, isto é, a quantidade de informação que existe em cada registro carregado.

Nesse cenário, é intuitivo que quanto maior o valor de m, menor será o valor da árvore e assim as operações de busca, inserção e remoção serão mais rápidas. Contudo, quanto maior m, maior o número de registros em cada nó - especialmente se a carga for alta - , o que tem maior custo, e assim há um relação de compromisso entre esses parâmetros.

Portanto, a metodologia consiste essencialmente em começar alterando o valor de m a partir de m = 2. A CARGA será variada de 10^4 a 10^6, aumentando sempre por um fator 10. Assim, é possível ver o impacto do aumento da CARGA na árvore com parâmetro m = 2. Feito isso, o parâmetro m será variado em fator 2 (2, 4, 8, 16, etc) até m = 64, de modo a ver o impacto que o parâmetro m tem no desempenho, bem como o impacto da CARGA nos valores de m.

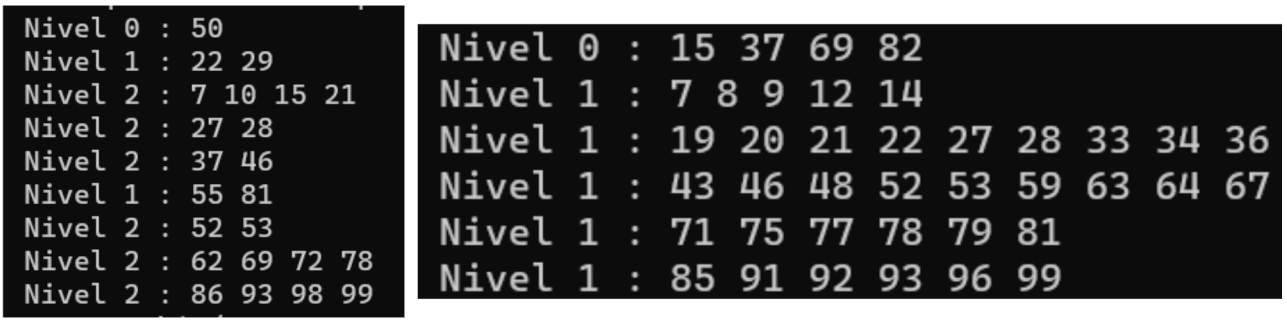
Será utilizado o utilitário time do linux, que já calcula o tempo de execução do binário. Os tempos gastos para cada configuração de parâmetros serão coletados e inseridos na Tabela 3.1, disponível na seção 3. Note que o código disponibilizado no moodle já efetua diversas operações de inserção, remoção e pesquisa na árvore, assim calcular o tempo total gasto pelo binário é suficiente para aferir o impacto dos parâmetros.

No makefile, será usado a seguinte linha para compilar, passando os argumentos de -Dm e -DCARGA que fornecem o valor dos parâmetros analisados.

```
$(CC) $(CFLAGS) -Dm=2 -DCARGA=1000 -o $(OBJ)/AP10.o $(SRC)/AP10.c
```

Esses parâmetros serão alterados a cada execução, sendo recalculado o tempo de execução chamando-se novamente o utilitário time do linux. O impacto do parâmetro m é visível quando setamos exp.print = 1 na main, de modo a ver o que está acontecendo no código, como mostra a Figura 2.1.

Figura 2.1. Árvore B simulada com parâmetro m igual a (a) 2 e (b) 5, com exp.print = 1.



(a) (b)

Fonte: elaboração própria.

Como mostrado na Figura 2.1, temos que o aumento do parâmetro m de 2 para 5 fez com que cada nó não somente tenha mais registros, como também tenha mais filhos. Nesse cenário, podemos concluir que o código está funcionando bem, efetivamente simulando uma árvore B com os parâmetros, e seguir com o experimento.

É importante destacar que a configuração exp.numops deve ser elevada (usamos 10^5), para que sejam executadas muitas operações sobre a árvore B, de modo a melhor aferir o impacto dos parâmetros sobre o tempo de execução;

3 RESULTADOS

A Tabela 3.1 exibe os tempos de execução do código para cada uma das configurações de parâmetros da árvore B.

Tabela 3.1. Tempos de execução do código para cada configuração de parâmetros da árvore B.

m	CARGA	Tempo de execução (ms)
2	1000	30
2	10000	70
2	100000	940
4	1000	20
4	10000	70
4	100000	1010
8	1000	30
8	10000	70
8	100000	1140
16	1000	30
16	10000	100
16	100000	Erro
32	1000	40
32	10000	140
32	100000	Erro
64	1000	40
64	10000	150
64	100000	Erro

Fonte: elaboração própria.

Analisando os dados coletados na Tabela 3.1, temos que o aumento do parâmetro m na verdade aumentou os tempos de execução do código, ao contrário do esperado pela nossa intuição. Nesse contexto, vemos que apesar de as árvores serem menores (devido ao maior m), o aumento do número de registros em cada nó fica maior, de tal modo que o tempo total gasto das operações aumenta. Isso também ocorre com maiores valores do parâmetro CARGA.

A partir de $m = 16$ e com a CARGA igual a 10^5, não conseguimos extrair o tempo de execução do binário uma vez que o código retorna erro de compilação. Tentamos

debugar o código disponibilizado no moodle com gdb, mas não consegui resolver o problema até a data de entrega da prática. No entanto, a partir das medições para $m \leq 16$ anteriores na Tabela 3.1, podemos inferir que o tempo gasto para carga 10^5 com m igual a 16, 32 e 64 seria elevado, sendo superior a 1000 ms.

4 CONCLUSÃO

Tendo em vista o objetivo da prática, foi possível verificar o impacto que os parâmetros CARGA e m tem uma árvore B simulada, apesar de erros pontuais encontrados no código durante alguns experimentos.