

**Aluno:** Jhonatan Guilherme de Oliveira Cunha

**RA:** 2135590

**Disciplina:** Algoritmo e Estrutura de Dados 2

**6) Preencha a tabela abaixo usando as funções que você implementou. A tabela mostra a quantidade de colisões que aconteceram com cada função de sondagem. Para cada N, crie uma tabela hash com tamanho inicial  $M = 5$  e insira as chaves geradas aleatoriamente com a função `int* random_vector(int n, int max, int seed)`. Use  $\text{max} = 10 \cdot n$  e  $\text{seed} = 42$ .**

COLISÕES		N				
		100	1000	10000	100000	1000000
FUNÇÃO DE SONDAGEM	LINEAR	6	155	2996	11121	184337
	QUADRÁTICO	6	151	2788	10745	176637
	DUPLO	7	152	2456	10329	167416

### **7. Analisando os resultados obtidos no item 6:**

**I. Alguma função de sondagem resultou em consistentemente menos colisões para todos os N analisados? Se sim, qual?**

**R:** Sim, no geral a função de hash duplo resultou em menos colisões durante a execução do algoritmo.

**II. Como que os números de colisões se comparam com os N correspondentes? O que isso quer dizer para o custo da busca? O que aconteceria se permitíssemos que  $\alpha$  ficasse maior antes de redimensionar a tabela?**

**R:** Os números de colisões são bem inferiores ao seu N correspondente, isto nos mostra que o fator de carga de nossa tabela hash sempre será inferior a  $\frac{1}{2}$ . Caso nosso fator de carga for maior que  $\frac{1}{2}$ , não teremos convicção que nosso elemento poderá ser inserido na tabela.

**III. Quais outros padrões você identificou? Explique.**

**R:** Podemos notar que a função de sondagem linear apresenta bem mais colisões que as outras duas. Isto se deve a forma como a função de hash foi implementado, formando assim *clusters* maiores.