4.4 Tarefa 4

A Treap teve o melhor desempenho de todas as árvores. A sua natureza aleatória permite inserir elementos com um número mínimo de rotações. O balanceamento é o menos restrito de todas as árvores logo as pesquisas vão ser mais demoradas em média do que nas árvores AVL e Red Black.

Nota também que para o conjuntos A e B, os quais são ordenados, o desempenho é significativamente menor que para os conjuntos aleatórios C e D.

Anexo B - Código de Autor

```
/* Código feito para C99, compilado com GCC 14.2.1 com flags --std=c99 -O2 e --fast-math
 *
 * Hardware Original: TOSHIBA SATELLITE_C50-A PSCG6P-01YAR1,
 * CPU: Intel i5-3320M (4) @ 3.300GHz,
 * GPU: Intel 3rd Gen Core processor Graphics Controller
 * RAM: 7821MiB, SSD SATA3 1TB
 *
 * Aluno: Vasco Alves, 2022228207
*/

#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include <assert.h>
#include <time.h>

#define RESIZE_FACTOR 1.61803

#define IDX_INVALID 4294967295

#define SEED 95911405

#define BLACK 0
#define RED 1

typedef uint32_t idx_t;
typedef int32_t key_t;

static int32_t g_treesize;
static int32_t g_average;
static uint32_t g_rotation_count = 0;

typedef struct BinTreeNode {
```