multithreaded/hash.h

```c
/*
A hash set with strings as keys
- uses Linux's hsearch as the underlying hashmap
*/

#define _GNU_SOURCE
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <search.h>
#include "p_stack.h"

typedef struct hashmap
{
    // current capacity (max number of keys)
    size_t size;
    // array of keys in the hash set
    //  - for linear access,
    //  - for moving keys over on resize,
    //  - and for deallocation at destruction
    char **elements;
    // hsearch structure
    struct hsearch_data *hmap;
    // current number of keys
    size_t cur_size;
    // stack of pointers to strings used for searching in hsearch;
    //  so we can deallocate them at cleanup
    PSTACK *ps;
} HSET;

#define HSET_RESIZE_FACTOR 2

int init_hset(HSET *p, size_t set_size);
bool is_full_hset(HSET *p);
bool is_empty_hset(HSET *p);
int add_hset(HSET *p, char *key);
int search_hset(HSET *p, char *key);
int resize_hset(HSET *p);
int cleanup_hset(HSET *p);
```