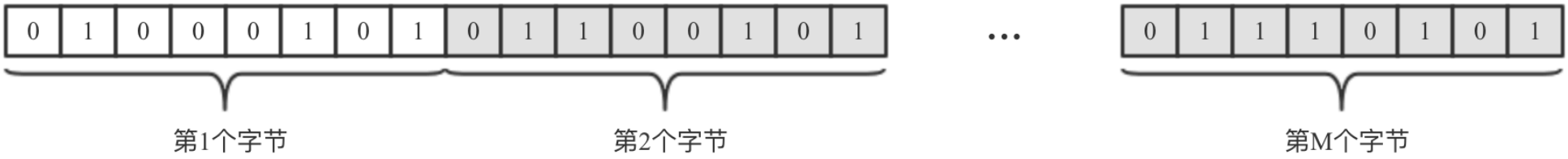
需要哪些组件单元

1. 判定格子是否加入过堆中
2. 堆

位图索引



位运算：

1) 设置第i位占用
2) 判定第i位是否占用

位图索引

```
#ifndef SERVER_GRID_LBS_BITMAP_H_
#define SERVER_GRID_LBS_BITMAP_H_

#include <stdint.h>
#include <stdlib.h>

typedef struct lbs_bitmap_s {
  uint8_t *bits;
  uint32_t bits_num;
} lbs_bitmap_t;

/** 初始化Bitmap **/
int lbs_bitmap_init(lbs_bitmap_t* lbs_bitmap, uint32_t bits_num);
/** 销毁 **/
int lbs_bitmap_destroy(lbs_bitmap_t* lbs_bitmap);
/** 设置Bit**/
int lbs_bitmap_setbit(lbs_bitmap_t* lbs_bitmap, uint32_t pos);
/** 取消设置Bit **/
int lbs_bitmap_unsetbit(lbs_bitmap_t* lbs_bitmap, uint32_t pos);
/** 判定是否设置Bit **/
int lbs_bitmap_isset(lbs_bitmap_t* lbs_bitmap, uint32_t pos);

#endif  // SERVER_GRID_BITMAP_H_
```

堆结构

```
#ifndef SERVER_GRID_LBS_NN_HEAP_H_
#define SERVER_GRID_LBS_NN_HEAP_H_

#include "server/grid/lbs_defs.h"

typedef struct lbs_heapnode_s {
  double distance;    // 距离
  uint8_t is_grid;    // 1:是网格   0:移动对象
  int cell_id;     // cell id
  lbs_mov_node_t* node;
} lbs_heapnode_t;

typedef struct lbs_nnheap_s {
  uint32_t capacity;
  uint32_t size;
  lbs_heapnode_t *heap_nodes;
} lbs_nnheap_t;

/** 初始化 **/
int lbs_nnheap_init(lbs_nnheap_t* lbs_nnheap);
/** 销毁 **/
int lbs_nnheap_destroy(lbs_nnheap_t* lbs_nnheap);
/** 插入 **/
int lbs_nnheap_insert(lbs_nnheap_t* lbs_nnheap,
                      lbs_mov_node_t* lbs_mov_node,
                      int cell_id, uint8_t is_grid, double distance);
/** 获取离distance最小的lbs_heapnode_t ***/
lbs_heapnode_t* lbs_nnheap_top(lbs_nnheap_t* lbs_nnheap);
/** 删除堆顶元素 ***/
void lbs_nnheap_pop(lbs_nnheap_t* lbs_nnheap);

#endif  // SERVER_GRID_LBS_NN_HEAP_H_
```