로고, 등록 상표, 폰트, 상징이(가) 표시된 사진

자동 생성된 설명

**데이터구조2**

**과제#1-1 minheap**

|  |  |
| --- | --- |
| 과목명 | 데이터구조 |
| 담당교수 | 남윤영 |
| 학과 | 컴퓨터공학과 |
| 학년 | 2 |
| 학번 | 20204101 |
| 이름 | 신동욱 |
| 제출일 | 09.12 |

**1. 소스코드**

/\*

20204101 신동욱

과제#1-1 minheap

\*/

#include <stdio.h>

#include <stdlib.h>

#define MAX\_ELEMENT 200

typedef struct {

    int key;

}element;

typedef struct {

    element heap[MAX\_ELEMENT];

    int heap\_size;

}HeapType;

HeapType\* create() {

    return (HeapType\*)malloc(sizeof(HeapType));

}

void init(HeapType\* h) {

    h->heap\_size = 0;

}

void insert\_min\_heap(HeapType\* h, element item) {

    int i;

    i = ++(h->heap\_size);

    while ((i != 1) && (item.key < h->heap[i / 2].key)) {

        h->heap[i] = h->heap[i / 2];

        i /= 2;

    }

    h->heap[i] = item;

}

element delete\_min\_heap(HeapType\* h) {

    int parent, child;

    element item, temp;

    item = h->heap[1];

    temp = h->heap[(h->heap\_size)--];

    parent = 1;

    child = 2;

    while (child <= h->heap\_size) {

        if ((child < h->heap\_size) && (h->heap[child].key) > h->heap[child + 1].key)

            child++;

        if (temp.key < h->heap[child].key)

            break;

        h->heap[parent] = h->heap[child];

        parent = child;

        child \*= 2;

    }

    h->heap[parent] = temp;

    return item;

}

int main() {

    HeapType\* heap;

    heap = create();

    init(heap);

    int a[] = { 10, 40, 30, 5, 12, 6, 15, 9, 60 };

    element e;

    for (int i = 0; i < 9; i++) {

        e.key = a[i];

        insert\_min\_heap(heap, e);

    }

    element d;

    for (int i = 0; i < 9; i++) {

        d = delete\_min\_heap(heap);

        printf("%d ", d.key);

    }

}

텍스트, 폰트, 도표이(가) 표시된 사진

자동 생성된 설명**2. 실행 과정** insert\_min\_heap

텍스트, 도표, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명delete\_min\_heap

**3. 실행 결과**



크기가 작은 순서대로 출력됨

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

가시성을 위해 히프의 상태를 출력한 결과

(별도의 print\_heap   
함수를 작성하였습니다.)