

CS2351 DATA STRUCTURES

Binary Heap

Heap

```
#include <stdio.h>

#define MAX_SIZE 100
int count = 0;
int H[MAX_SIZE+1];

int insert(int *heap, int value) {
    int parent, current, temp;
    count++;
    if ( count >= MAX_SIZE ) { printf("Heap overflow!\n"); return -1; }
    heap[count] = value;

    current = count; parent = current/2 ;
    while ( parent >= 1 && heap[current] < heap[parent] ) {
        temp = heap[current]; heap[current] = heap[parent]; heap[parent] = temp;
        current = parent; parent = current/2 ;
    }
    return count;
}

void extract_min(int *heap) {
    int current, left_child, right_child, min_child, temp;

    heap[1] = heap[count]; count--;
    current = 1; left_child = 2; right_child = 3;

    while ( left_child <= count ) {
        if ( right_child > count || heap[left_child] < heap[right_child] )
            min_child = left_child;
        else min_child = right_child;

        if ( heap[current] > heap[min_child] ) {
            temp = heap[current]; heap[current] = heap[min_child]; heap[min_child] = temp;
            current = min_child; left_child = 2 * current; right_child = 2 * current + 1;
        }
        else break;
    }
}

int find_min(int *heap) {
    return heap[1];
}
```

```
int main() {
    int B[8] = { 18, 3, 5, 1, 12, 17, 8, 2 }, i, j;

    for (i = 0; i < 8; i++ )
        insert(H, B[i]);

    while ( count >= 1 ) {
        j = find_min(H); extract_min(H); printf("%d\n", j);
    }

    return 0;
}
```