Heap練習

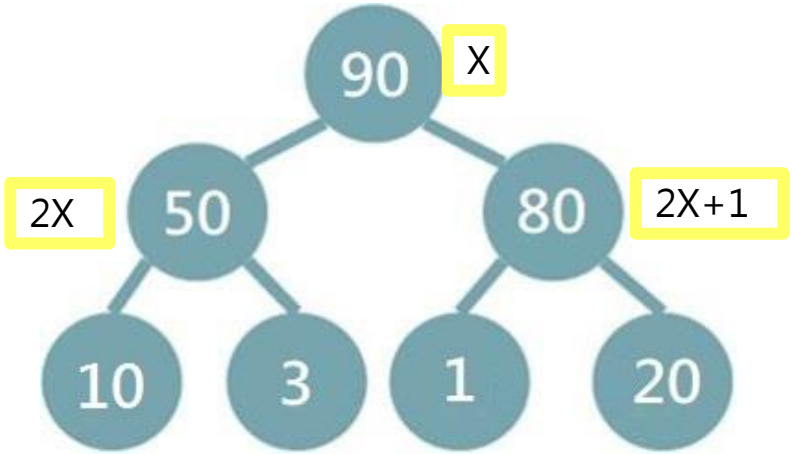
- 建立MaxHeap {10, 50, 20, 80, 3, 1, 90}
- 使用Heap進行排序

```
10, 50, 20, 80, 3, 1, 90,
90, 80, 50, 20, 10, 3, 1,
```

```
10, 50, 20, 80, 3, 1, 90, 11, 33, 81, 47,
90, 81, 80, 50, 47, 33, 20, 11, 10, 3, 1,
```

Heap練習

- 使用陣列建立Heap



{X, 90, 50, 80, 10, 3, 1, 20}

Heap練習

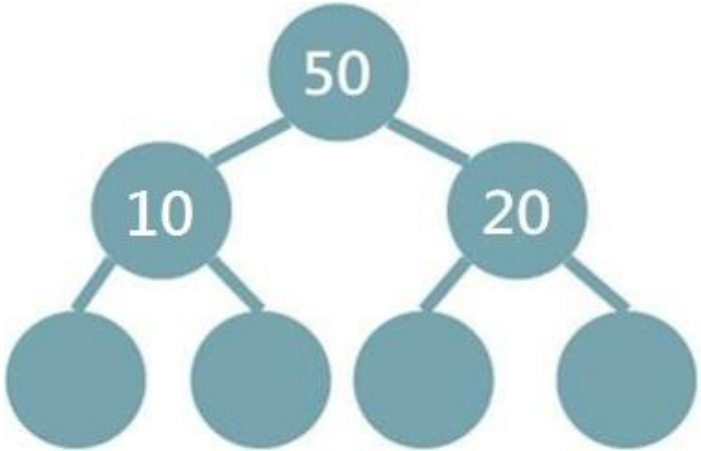
- 建立MaxHeap {10, 50, 20, 80, 3, 1, 90}

```
void insert_heap (int heap[], int value)
{
    int new_index = tail + 1;
    int parent;
    int temp;

    heap[new_index] = value;

    while (new_index > 1) {

        parent = new_index / 2;

        if (heap[new_index] > heap[parent]) {
            temp = heap[new_index];
            heap[new_index] = heap[parent];
            heap[parent] = temp;

            new_index = parent;

        } else
            break;

    }

    tail ++;
}
```

插入value

調整Heap

X	50	10	20				



VRMLAB
虛擬實境暨多媒體實驗室

Heap練習

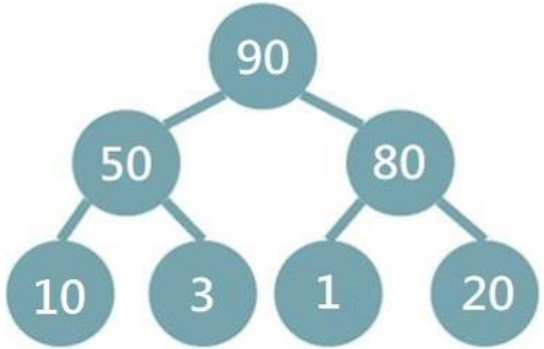
- 使用Heap進行排序

```
void delete_heap (int heap[])
{
    int check_index = 1;
    int child;
    int temp;

    printf ("%d, ", heap[1]);

    heap[1] = heap[tail];
    tail --;
```

刪除value

```
while (check_index * 2 <= tail) {

    child = check_index * 2;

    if (child + 1 <= tail) {
        if (heap[child] < heap[check_index] && heap[child + 1] < heap[check_index])
            break;

        if (child + 1 <= tail && heap[child + 1] > heap[child])
            child ++;

    } else {
        if (heap[child] < heap[check_index])
            break;
    }

    temp = heap[check_index];
    heap[check_index] = heap[child];
    heap[child] = temp;

    check_index = child;

}
```

調整Heap



VRMLAB
虛擬實境暨多媒體實驗室