

Heaps

A Introduction to Heaps

A heap is a specialized tree-based data structure that satisfies the heap property: in a max heap, for any given node, the value is greater than or equal to its children; in a min heap, the value is less than or equal to its children. Heaps are commonly used to implement priority queues and for efficient sorting (heapsort).

- **Complete Binary Tree:** All levels are fully filled except possibly the last, which is filled from left to right.
- **Heap Property:** Max heap ($\text{parent} \geq \text{children}$), min heap ($\text{parent} \leq \text{children}$).
- **Efficient Operations:** Insert, delete, and access max/min in $O(\log n)$ time.

B Applications

- Priority queues
- Heapsort algorithm
- Graph algorithms (Dijkstra's, Prim's)
- Scheduling systems

C C++ Example: Min Heap

Below is a simple C++ implementation of a min heap using a vector.

```
#include <iostream>
#include <vector>
using namespace std;

class MinHeap {
    vector<int> heap;
    void heapifyUp(int i) {
        while (i > 0 && heap[(i-1)/2] > heap[i]) {
            swap(heap[i], heap[(i-1)/2]);
            i = (i-1)/2;
        }
    }
    void heapifyDown(int i) {
        int n = heap.size();
        while (2*i+1 < n) {
            int j = 2*i+1;
            if (j+1 < n && heap[j+1] < heap[j]) j++;
            if (heap[i] <= heap[j]) break;
            swap(heap[i], heap[j]);
            i = j;
        }
    }
public:
    void insert(int val) {
        heap.push_back(val);
        heapifyUp(heap.size()-1);
    }
    int extractMin() {
        int minVal = heap[0];
        heap[0] = heap.back();
```

```

        heap.pop_back();
        heapifyDown(0);
        return minVal;
    }
    void print() {
        for (int v : heap) cout << v << " ";
        cout << endl;
    }
};

int main() {
    MinHeap h;
    h.insert(5); h.insert(3); h.insert(8); h.insert(1);
    h.print();
    cout << "Extracted min: " << h.extractMin() << endl;
    h.print();
    return 0;
}

```

D Practice

Practice Exercise:

Implement a max heap in C++. Add methods for insertion, extraction of the maximum, and printing the heap. Test your implementation with a sequence of insertions and extractions.