

```
// Raul Martinez
// raul.martinez01@utrgv.edu

#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

template <class T>
class minHeap
{
public:
    vector<T> item;

    minHeap()
    {
        // do nothing
    }

    int parent(int i)
    {
        if (i == 0)
            return i;

        else if (i % 2 == 0)
            return (i - 1) / 2;

        else
            return i / 2;
    }

    int lchild(int i)
    {
        return i + (i + 1);
    }

    int rchild(int i)
    {
        return i + (i + 2);
    }

    void insert(T x)
    {
        item.push_back(x);
        int last = item.size() - 1;
        bubbleUp(last);
    }

    T extractMin()
    {
        T val = item[0];

        if (item.empty())
        {
            // do nothing
        }
    }
};
```

```
    }
    else if (item.size() == 1)
    {
        item.pop_back();
        return val;
    }
    else
    {
        swap(item[item.size() - 1], item[0]);
        item.pop_back();
        bubbleDown(0);
        return val;
    }

    return val;
}

void bubbleUp(int i)
{
    if (item[parent(i)]->weight > item[i]->weight)
    {
        item[i]->index = parent(i);
        item[parent(i)]->index = i;
        swap(item[i], item[parent(i)]);
        bubbleUp(parent(i));
    }
    else
    {
        item[i]->index = i;
    }
}

void bubbleDown(int i)
{
    int l = lchild(i);
    int r = rchild(i);

    int minI = i;

    if (l < item.size() && item[l]->weight < item[i]->weight)
    {
        item[i]->index = l;
        item[l]->index = i;
        minI = l;
    }
    if (r < item.size() && item[r]->weight < item[minI]->weight)
    {
        item[i]->index = r;
        item[r]->index = i;
        minI = r;
    }
    if (minI != i)
    {
        swap(item[i], item[minI]);
        bubbleDown(minI);
    }
}
```

```
    }

    void display()
    {
        for (int i = 0; i < item.size(); i++)
            cout << "Index: " << i << ": " << item[i] << endl;
    }

    bool empty()
    {
        return item.size() == 0;
    }
};
```