```cpp
int parent(int i) {
    return (i - 1) / 2;
}

int left(int i) {
    return 2 * i + 1;
}

int right(int i) {
    return 2 * i + 2;
}

void shiftUp(int a[], int n, int i) {
    while (i > 0 && a[i] > a[parent(i)]) {
        swap(a[i], a[parent(i)]);
        i = parent(i);
    }
}

void shiftDown(int a[], int n, int i) {
    int max_index = i;
    while (1) {
        if (left(i) < n && a[left(i)] > a[max_index]) {
            max_index = left(i);
        }
        if (right(i) < n && a[right(i)] > a[max_index]) {
            max_index = right(i);
        }
        if (max_index == i) {
            return;
        }
        swap(a[i], a[max_index]);
        i = max_index;
    }
}

int extract(int a[], int& n) {
    int res = a[0];
    a[0] = a[n - 1];
    n--;
    shiftDown(a, n, 0);
    return res;
}

void insert(int a[], int& n, int k) {
    n++;
    a[n - 1] = k;
    shiftUp(a, n, n - 1);
}
```

```cpp

void changePriority(int a[], int n, int i, int new_prior) {
    int old_prior = a[i];
    a[i] = new_prior;
    if (new_prior > old_prior) {
        shiftUp(a, n, i);
    }
    else {
        shiftDown(a, n, i);
    }
}

void remove(int a[], int& n, int i) {
    a[i] = a[0] + 1;
    shiftUp(a, n, i);
    extract(a, n);
}
```