

```
1 int parent(int i) {
2     return (i - 1) / 2;
3 }
4
5 int left(int i) {
6     return 2 * i + 1;
7 }
8
9 int right(int i) {
10    return 2 * i + 2;
11 }
12
13 void shiftUp(int a[], int n, int i) {
14     while (i > 0 && a[i] > a[parent(i)]) {
15         swap(a[i], a[parent(i)]);
16         i = parent(i);
17     }
18 }
19
20 void shiftDown(int a[], int n, int i) {
21     int max_index = i;
22     while (1) {
23         if (left(i) < n && a[left(i)] > a[max_index]) {
24             max_index = left(i);
25         }
26         if (right(i) < n && a[right(i)] > a[max_index]) {
27             max_index = right(i);
28         }
29         if (max_index == i) {
30             return;
31         }
32         swap(a[i], a[max_index]);
33         i = max_index;
34     }
35 }
36
37 int extract(int a[], int& n) {
38     int res = a[0];
39     a[0] = a[n - 1];
40     n--;
41     shiftDown(a, n, 0);
42     return res;
43 }
44
45 void insert(int a[], int& n, int k) {
46     n++;
47     a[n - 1] = k;
48     shiftUp(a, n, n - 1);
49 }
```

```
50
51 void changePriority(int a[], int n, int i, int new_prior) {
52     int old_prior = a[i];
53     a[i] = new_prior;
54     if (new_prior > old_prior) {
55         shiftUp(a, n, i);
56     }
57     else {
58         shiftDown(a, n, i);
59     }
60 }
61
62 void remove(int a[], int& n, int i) {
63     a[i] = a[0] + 1;
64     shiftUp(a, n, i);
65     extract(a, n);
66 }
```