05/24/17 01:58:13 /Users/hostname/Desktop/CSE 330/Lab7/heap_linkedlist.cpp

```cpp
// Adnar Lozano
// CSE 330 Data Structures
// Lab 7
// 5/20/17

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

void rebuild(vector<int> &heap, int& cur_size);

void addItem(vector<int> &heap, int item, int& cur_size) {
    if (cur_size > 0) {
        cur_size++;
        heap.push_back(item);
        int index = cur_size - 1;
        while ((index-1)/2 >= 0) {
            int parent = (index - 1) / 2;
            if (heap.at(index) < heap.at(parent)) {
                int tmp = heap.at(parent);
                heap.at(parent) = heap.at(index);
                heap.at(index) = tmp;
                index = parent;
            }
            else
                return;
        }
    }
    else if (cur_size == 0) {
        cur_size++;
        heap.push_back(item);
    }
    else
        return;
}

void deleteItem(vector<int> &heap, int item, int& cur_size) {
    int index = 0;
    while (heap.at(index) != item)
        index++;
    heap.at(index) = heap.back();
    heap.resize(--cur_size);
    while (index > 0) {
        if (2 * index + 1 < cur_size) {
            int left = 2 * index + 1;
            int right = left + 1;
            int swap;
            if (right < cur_size) {
                if (heap.at(left) < heap.at(right))
                    swap = left;
                else if (heap.at(left) > heap.at(right))
                    swap = right;
                else
                    swap = left;
            }
            else
                swap = left;

            if (heap.at(index) > heap.at(swap)) {
                int tmp = heap.at(index);
                heap.at(index) = heap.at(swap);
                heap.at(swap) = tmp;
            }
        }

        index = (index - 1) / 2;
    }
    rebuild(heap, cur_size);
```

```
69  }
70
71  void rebuild(vector<int> &heap, int &cur_size) {
72      int index = 0;
73      while (2 * index + 1 < cur_size) {
74          int left = 2 * index + 1;
75          int right = left + 1;
76          int swap;
77          if (right < cur_size) {
78              if (heap.at(left) < heap.at(right))
79                  swap = left;
80              else if (heap.at(left) > heap.at(right))
81                  swap = right;
82              else
83                  swap = left;
84          }
85          else
86              swap = left;
87          if (heap.at(index) > heap.at(swap)) {
88              int tmp = heap.at(index);
89              heap.at(index) = heap.at(swap);
90              heap.at(swap) = tmp;
91          }
92          index = swap;
93      }
94  }
95
96  int main() {
97      int opCount;
98      int cur_size = 0;
99      vector<int> heap;
100     cin >> opCount;
101     while (opCount>0) {
102         int opID;
103         cin >> opID;
104         if (opID == 1) {
105             int item;
106             cin >> item;
107             addItem(heap, item, cur_size);
108         }
109         else if (opID == 2) {
110             int item;
111             cin >> item;
112             deleteItem(heap, item, cur_size);
113         }
114         else if (opID == 3)
115             cout << heap.at(0) << endl;
116         else
117             break;
118         opCount--;
119     }
120     return 0;
121 }
```