

## Execução

Criei as funções heapify para cima e para baixo que serão usadas para fazer o heap depois de adicionar ou deletar um elemento e as funções de adicionar e remover um elemento, e para facilitar a main criei também o exibir heap. Na main eu gerei dados aleatórios para preencher o vetor e fazer os testes de remover e adicionar novos dados sempre mostrando a cada passo como o vetor fica.

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  #include <ctime>
5  #include <cstdlib>
6
7
8  using namespace std;
9
10 void heapifyParaCima(vector<char>& heap, int indice) {
11     if (indice == 0)
12         return;
13
14     int indicePai = (indice - 1) / 2;
15     if (heap[indice] > heap[indicePai]) {
16         swap(heap[indice], heap[indicePai]);
17         heapifyParaCima(heap, indicePai);
18     }
19 }
20
21 void heapifyParaBaixo(vector<char>& heap, int indice, int tamanho) {
22     int filhoEsquerdo = 2 * indice + 1;
23     int filhoDireito = 2 * indice + 2;
24     int maior = indice;
25
26     if (filhoEsquerdo < tamanho && heap[filhoEsquerdo] > heap[maior])
27         maior = filhoEsquerdo;
28
29     if (filhoDireito < tamanho && heap[filhoDireito] > heap[maior])
30         maior = filhoDireito;
```

```
31
32     if (maior != indice) {
33         swap(heap[indice], heap[maior]);
34         heapifyParaBaixo(heap, maior, tamanho);
35     }
36 }
37
38 void construirHeapDeBaixoParaCima(vector<char>& heap) {
39     int tamanho = heap.size();
40     for (int i = tamanho / 2 - 1; i >= 0; --i) {
41         heapifyParaBaixo(heap, i, tamanho);
42     }
43 }
44
45 void adicionarElemento(vector<char>& heap, char valor) {
46     heap.push_back(valor);
47     heapifyParaCima(heap, heap.size() - 1);
48 }
49
50 void removerElemento(vector<char>& heap) {
51     if (heap.empty())
52         return;
53
54     heap[0] = heap.back();
55     heap.pop_back();
56     heapifyParaBaixo(heap, 0, heap.size());
57 }
58
59 void exibirHeap(const vector<char>& heap) {
60     for (char valor : heap) {
```

```

61         cout << valor << " ";
62     }
63     cout << endl;
64 }
65
66 int main() {
67     srand(time(0));
68
69     vector<char> dados = {'E', 'A', 'C', 'D', 'B'};
70     construirHeapDeBaixoParaCima(dados);
71
72     cout << "Vetor Heap : ";
73     exibirHeap(dados);
74
75     removerElemento(dados);
76     cout << "Heap remocao de um elemnto: ";
77     exibirHeap(dados);
78
79     char novoElemento = 'F';
80     adicionarElemento(dados, novoElemento);
81     cout << "Heap adicao de um elemento " << novoElemento << ": ";
82     exibirHeap(dados);
83
84     // Remocao e adicao de mais elementos
85     removerElemento(dados);
86     cout << "Heap remocao de mais um elemnto: ";
87     exibirHeap(dados);
88
89     char novoElementoAleatorio = 'A' + rand() % 26; // Gera letra aleatoria entre 'A' e 'Z'
90     adicionarElemento(dados, novoElementoAleatorio);

```

## Dificuldades

Tive menos dificuldade nesse do que no outro exercício já que este tinha uma base nos slides, foi entender como usar adequadamente o construir de baixo pra cima e o de cima pra baixo e eu resolvi o problema vendo a vídeo aula do professor disponível no youtube sobre a matéria.

## Resultado

```

C:\WINDOWS\system32\cmd.exe
Vetor Heap : E D C A B
Heap remocao de um elemnto: D B C A
Heap adicao de um elemento F: F D C A B
Heap remocao de mais um elemnto: D B C A
Heap adicao de mais um elemento : U D C A B

Pressione qualquer tecla para continuar. . .

```