A ideia é construir uma heap máxima, isto é, uma árvore tal que todo nó pai é necessariamente maior que ambos os nós filhos, dessa forma, se a heap for máxima vc sabe que a primeira raiz deve ir para o fim do vetor.

Após isso vc vai trocar quem tava na primeira raiz com quem estava no último lugar do vetor e tirar esse cara da heap. Fazendo isso a sua heap deixa de ser heap máxima e vc vai rearranjar a heap novamente. É só fazer isso até que todos elementos já estejam ordenados.

O loop para criar a heap maxima começa em n/2, pois nem tem como vc criar uma heap com um número maior que n/2, pois todos eles não vão ter filhos.

Filho esq: 2\*i + 1

Filho dir: 2\*i + 2

Código:

void rearranjar(int \*v, int i, int n){

int esq = 2\*i + 1;

int dir = 2\*i + 2;

int maior = i;

if ((esq < n) && (v[esq] > v[maior])) maior = esq;

if ((dir < n) && (v[dir] > v[maior])) maior = dir;

if (maior == i) return;

swap(v, i, maior);

rearranjar(v, maior, n);

}

void heapsort(int \*v, int n){

//criar a heap

for (int i = n / 2; i >= 0; i--){

rearranjar(v, i, n);

}

for (int i = n - 1; i >= 0; i--){

swap(v, 0, i);

rearranjar(v, 0, i);

}

}

Complexidade: O(nlogn)