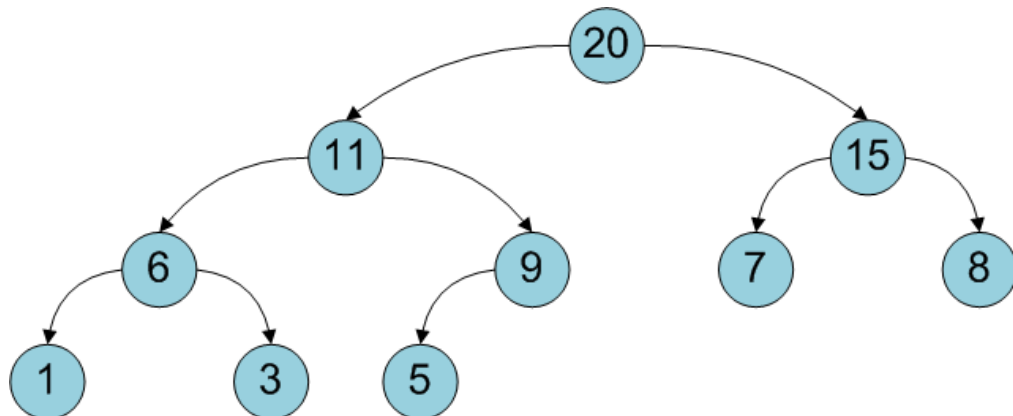


Пирамидальная сортировка (HeapSort)

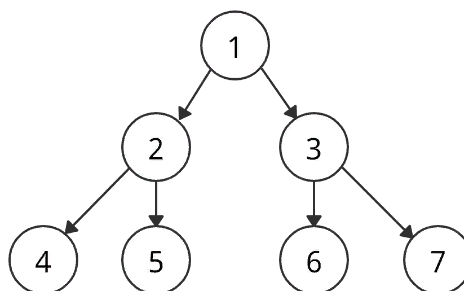
Пирамидальная сортировка (или сортировка кучей, HeapSort) — это метод сортировки сравнением, основанный на такой структуре данных как двоичная куча. Общая идея пирамидальной сортировки заключается в том, что сначала строится пирамида из элементов исходного массива, а затем осуществляется сортировка элементов.

Двоичная куча представляет собой законченное бинарное дерево, для которого выполняется основное свойство кучи: приоритет каждой вершины больше приоритетов её потомков. В простейшем случае приоритет каждой вершины можно считать равным её значению. (max-heap/min-heap)

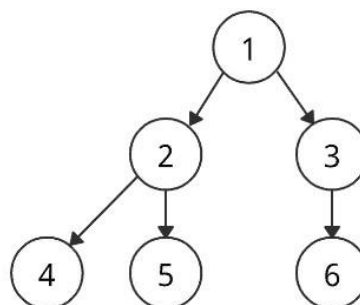


Законченное бинарное дерево

Законченное бинарное дерево отличается от совершенного несколькими критериями: все уровни должны быть заполнены, все вершины склоняются влево, у последней листовой вершины может не быть правого собрата – это значит, что завершённое дерево не обязательно должно быть совершенным.



Совершенное бинарное дерево



Законченное бинарное дерево

Поскольку двоичная куча — это законченное двоичное дерево, ее можно легко представить в виде массива, а представление на основе массива является эффективным с точки зрения расхода памяти. Если родительский узел хранится в индексе i , левый дочерний элемент может быть вычислен как $2i + 1$, а правый дочерний элемент — как $2i + 2$ (при условии, что индексирование начинается с 0).

Алгоритм пирамидальной сортировки

1. Постройте max-heap из входных данных.
2. На данном этапе самый большой элемент хранится в корне кучи. Замените его на последний элемент кучи, а затем уменьшите ее размер на 1. Наконец, преобразуйте полученное дерево в max-heap с новым корнем.
3. Повторяйте вышеуказанные шаги, пока размер кучи больше 1.

```
void in_heap(int* mas, int n, int _root){
    int root = _root;
    int l = 2 * _root + 1;
    int r = 2 * _root + 2;
    if (l < n && mas[l] > mas[root])
        root = l;
    if (r < n && mas[r] > mas[root])
        root = r;
    if (root != _root){
        swap(mas[_root], mas[root]);
        in_heap(mas, n, root);
    }
}

// Основная функция, выполняющая пирамидальную сортировку
void HeapSort(int* mas, int n){
    // Построение кучи (перегруппируем массив)
    for (int i = n / 2 - 1; i >= 0; i--){
        in_heap(mas, n, i);
    }
    // Один за другим извлекаем элементы из кучи
    for (int i = n - 1; i >= 0; i--){
        // Перемещаем текущий корень в конец
        swap(mas[0], mas[i]);
        // вызываем процедуру in_heap на уменьшенной куче
        in_heap(mas, i, 0);
    }
}
```

Временная сложность `in_heap` — $O(\log n)$. Временная сложность создания и построения кучи равна $O(n)$, а общее время работы пирамидальной сортировки — $O(n \log n)$.