

Неар - Пирамидално сортиране

Дървета

Множество от върхове, свързани с ребра, при което няма цикли.

Тръгвайки от дадено ребро, не може да се стигне до началния връх

Двоично дърво – дърво, на което всеки връх има най-много два наследника

Двоично дърво – състои се от корен, ляво поддърво и дясно поддърво (като е възможно поддърветата да са празни – без елементи). Връх, на който поддърветата са празни (т.е., няма **наследници**), наричаме **листо**.

Пълно двоично дърво (ПДД) – дърво, на което всеки връх има точно 2 наследника, като последното ниво от листа може да не е запълнено.

Пирамида (heap) – пълно двоично дърво, като стойността на даден връх е по-голяма или равна на стойностите на наследниците му – **свойство на пирамидата**. – В корена на пирамидата е разположен най-големият елемент (следствие от дефиницията)

Алгоритъм на пирамидалното сортиране

Функция Sift

Изисква се за връх i , който нарушава свойството на пирамидата. Като резултат функцията Sift прави пълното двоично поддърво, с корен връх i , на пирамида

l – индексът на върха, който ще се 'отсява'

r – индексът на последния елемент на пирамидата

```
public void Sift(int[] arr, int l, int r) {  
    int i = l, j = 2*i + 1;  
    int x = arr[i];  
    if(j < r && arr[j+1] > arr[j]) {  
        j++;  
    }  
}
```

```

while(j <= r && arr[j] > x) {
    arr[i] = arr[j];
    i = j;
    j = 2*i + 1;
    if(j < r && arr[j+1] > arr[j]) {
        j++;
    }
}
if(arr[i] != x) {
    arr[i] = x;
}
}

```

Функция HeapSort

```

public void HeapSort(int[] arr, int n) {
    for(int i = n/2; i >= 0; i--) {
        Sift(arr, i, n);
    }
    i = n-1;
    while (i > 0) {
        Swap(arr, 0, i);
        i--;
        Sift(arr, 0, i);
    }
}

```