

堆排序 (heap Sort)

1. 定义: 堆: n 个元素的序列 $k = (k_0, k_1, \dots, k_{n-1})$, 满足 $\begin{cases} k_i \geq k_{2i+1} \text{ 或 } k_i \geq k_{2i+2} \\ k_i \geq k_{2i+2} \end{cases}$ 或 $\begin{cases} k_i \leq k_{2i+1} \text{ 或 } k_i \leq k_{2i+2} \\ k_i \leq k_{2i+2} \end{cases}$

($i = 0, 1, \dots, \lfloor \frac{n}{2} \rfloor - 1$)

其中, 前一种叫大根堆, 后一种叫小根堆。

2. 算法: 创建初堆: 从第 $\lfloor \frac{n}{2} \rfloor - 1$ 号元素开始, 向上逐次调整堆。

去掉最大(小)元素重建堆: 每去掉一次堆顶元素, 就从 \bullet 至下调整堆一次。去顶操作以交换堆首尾元素实现。

3. 内核: (1) void heapSort (pvector)

```

{
  ① for (i = n/2 - 1; i >= 0; i--)          (创建初堆)
      sift (pvector, n, i);
  ② for (i = n - 1; i > 0; i--)          (重建堆)
      {
        temp = pvector -> record[0];
        pvector -> record[0] = pvector -> record[i];
        pvector -> record[i] = temp;
        sift (pvector, i, 0);
      }
}

```

Swap(pvector->record[0],
pvector->record[i]);
(交换函数)

(2) void sift (pvector, size, p)

{ while (child < size)

if ((child < size - 1) && (pvector->record[child].key > pvector->record[child + 1].key)

child++;

(找)

if (temp.key > pvector->record[child].key)

pvector->record[p] = pvector->record[child];

p = child; child = 2 * p + 1;

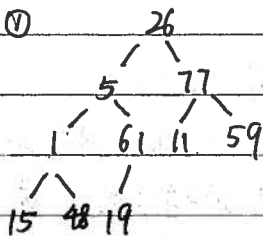
}

else break; (到位)

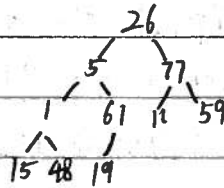
}

prevector → record[p] = temp;

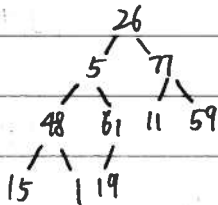
4. 图示: {26, 5, 77, 1, 61, 11, 59, 15, 48, 19} n=10, 大根堆



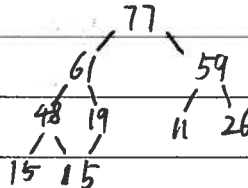
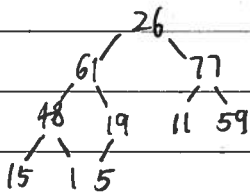
② $i = \frac{n}{2} - 1 = 4$, 对 61 调整 (不动)



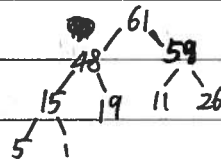
③ $i=3$, 对 1 调整 ($48 \leftrightarrow 1$) ④ $i=2$, 对 77 调整 (不动)



⑤ $i=1$, 对 5 调整 ($61 \leftrightarrow 5$) ($5 \leftrightarrow 19$) ⑥ $i=0$, 对 26 调整 ($26 \leftrightarrow 77$, $59 \leftrightarrow 26$)



初始化完成 / ⑦ 去 77, 调整 ($5 \leftrightarrow 61$, $5 \leftrightarrow 48$, $5 \leftrightarrow 15$)



⑧ 去 61, 调整 ($1 \leftrightarrow 59$, $1 \leftrightarrow 26$)

⑨ 去 59, 调整 ($5 \leftrightarrow 48$, $5 \leftrightarrow 19$)

⑩ 去 48, 调整 ($1 \leftrightarrow 26$, $1 \leftrightarrow 11$)

⑪ 去 26, 调整 ($1 \leftrightarrow 19$, $1 \leftrightarrow 15$)

⑫ 去 19, 调整 ($5 \leftrightarrow 15$) ⑬ 去 15, 调整 ($1 \leftrightarrow 11$) ⑭ 去 11, 调整 ($1 \leftrightarrow 5$)

⑮ 去 5.

空 $O(1)$ 时 $O(n \log n)$, 不稳定.