

# Homework4

刘喆骐 2020013163 探微化01

## 6-2

**a**

堆顶元素储存在下标为0处，对于下标为*i*的父节点，其*d*个子节点储存在下标为*di+1*, *di+2*, ..., *di+d*处。

**b**

对于深度为*h*的*d*叉树，其元素数量*n*满足如下关系：

$$\frac{d^h - 1}{d - 1} < n \leq \frac{d^{h+1} - 1}{d - 1}$$

$$\text{故 } h = \lceil \log_d(n(d - 1) + 1) - 1 \rceil$$

**c**

```
MAX-HEAPIFY(A,i)
    if id+1>A.heapsize-1 return
    temp=i
    for j in [id+1,id+d]
        if A[j]>A[temp] temp=j
    if temp != i
        swap(A[i],A[temp])
        MAX-HEAPIFY(A,temp)
```

```
EXTRACT-MAX(A)
    if A.heapsize<1:
        error "heap underflow"
    max=A[0]
    A[0]=A[A.heapsize-1]
    A.heapsize--
    MAX-HEAPIFY(A,0)
    return max
```

对于每一层的MAX-HEAPIFY，时间复杂度为 $O(d)$ ，最坏情况下需要进行到根节点，树高为 $O(\log_d n)$ ，故总体时间复杂度为 $O(d \log_d n)$

**d**

```
INSERT(A, key)
    A.heapsize++
    A[A.heapsize-1] = -inf
    INCREASE_KEY(A, A.heapsize, key)
```

其中INCREASE\_KEY会在e中实现，其时间复杂度为 $O(\log_d n)$ 。故总时间复杂度为 $O(\log_d n)$ 。

**e**

```
INCREASE_KEY(A, i, key)
    if key < A[i]
        error "new key is smaller"
    A[i] = key
    while A[i] > A[(i-1)/d] and i > 0
        swap(A[i], A[(i-1)/d])
        i = (i-1)/d
```

对于最坏情况，需要上滤至根节点，故时间复杂度为 $O(\log_d n)$

## 7-2

**a**

对于相同元素的数组，每次partition都会使得数组被分成了长度为1，n-1的两段，故时间复杂度为 $\Theta(n^2)$

**b**

```
PARTITION'(A, p, r)
  x = A[r]
  q = p
  t = p
  for j = p to r
    if A[j] < x
      y = A[j]
      A[j] = A[t + 1]
      A[t + 1] = A[q]
      A[q] = y
      q = q + 1
      t = t + 1
    else if A[j] == x
      swap(A[t + 1], A[j])
      t = t + 1
  return q, t
```

**c**

```
RANDOMIZED_QUICKSORT'(A, p, r)
  if p < r
    i = RANDOM(p, r)
    swap(A[i], A[r])
    q, t = PARTITION'(A, p, r)
    RANDOMIZED_QUICKSORT'(A, p, q - 1)
    RANDOMIZED_QUICKSORT'(A, t + 1, r)
```

**d**

由于我们不再比较和主元相同的元素，故QUICKSORT'的规模不会超过拥有完全不同元素的QUICKSORT的规模。对于 $i, j$ ，如果 $A[i] \neq A[j]$ ，两元素比较过的概率不变。而对于 $A[i] = A[j]$ ，假设 $A[i]$ 出现了 $x$ 次，则这两个元素比较过的概率是 $\frac{x-1}{C_x^2}$ ，而因为 $j - i + 1 \leq x$ ，因此新的比较次数 $\leq$ 原来的比较次数，结论不变。