1. i代表数组index（从0到n-1）

根节点：Heap[0]

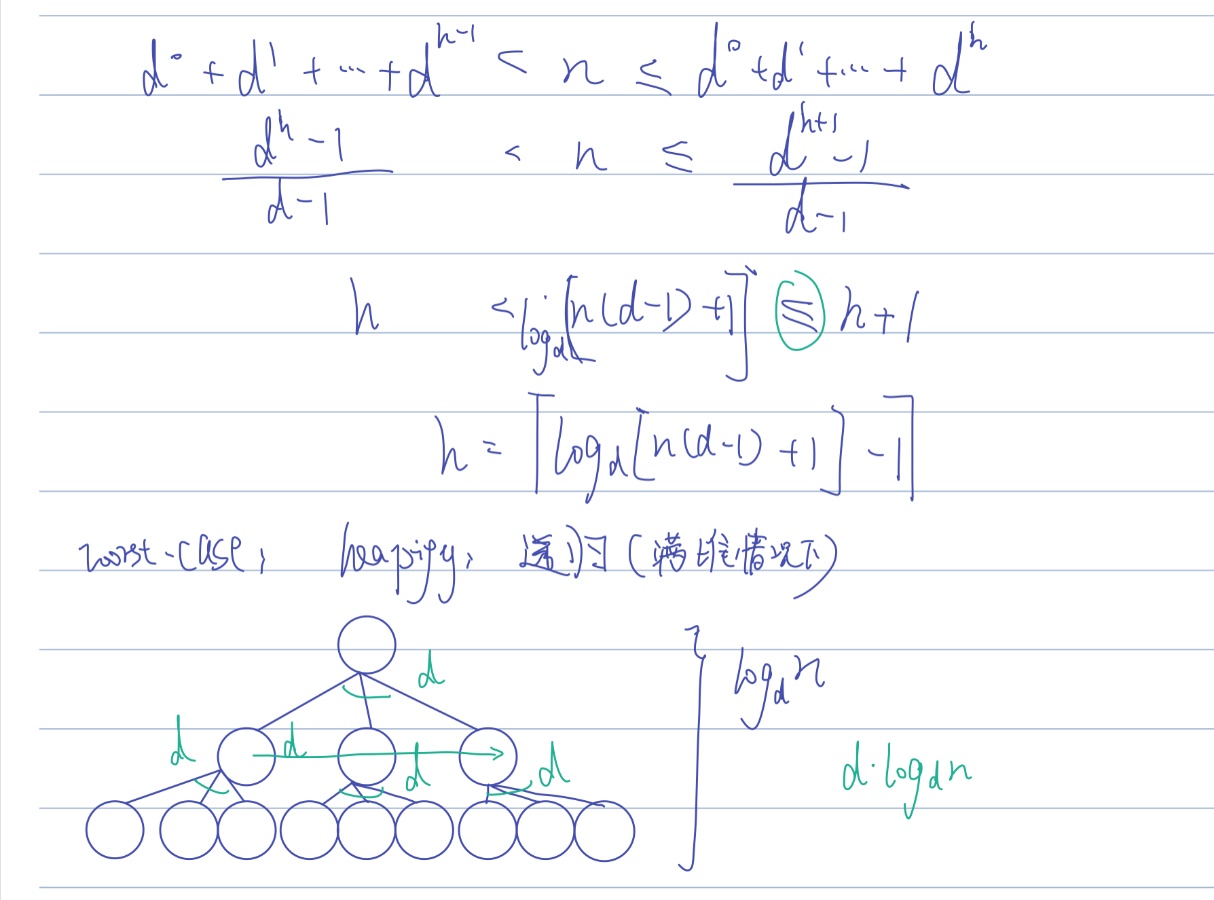
对于Heap[i]:（i>=0）

parent(i)（根节点无parent，i>0）

return (i-1)/d(向下取整)

k-th child (i,k)

return d\*i+k (k 从1到d)

2、height:

3.extract-max() 递归树分析运行时间见上图。树高logdn，最坏情况满堆且max和min互换。Min到达树顶后在每一层，循环d次选出最大的childnode和parentnode交换，若parentnode大于所有childnode则停止比较。最坏情况下每次都交换，一共 logdn层。所以时间复杂度O（dlogdn）

4.insert（）复用increase-key（）。

T（insert）: O(1)+T（increase-key）:O(logdn)

5.increase-key（）:

从下至上比较，若childnode值大于parentnode则二者交换，否则停止。worst-case:key大于原堆中的最大值，比较一直进行到根节点。比较次数等于树高：O(logdn)

代码中N代表静态分配的数组大小。HeapSize代表堆的大小，D代表D-叉堆