6.1 d

6.2 a

6.3 思路：整体思路与k-select相似，要选择一个分割点xi，递归调用权重较大的子数组继续求解即可。

伪代码：

**getSum**(x[], w[], start, end):

result = 0

for i in range(strat, end+1):

result += w[i]

return result

**WeightedMedian**(x[], w[], start, end):

mid = DSelect(x[], start, end)

L, pivotVal, R = Partion(x[], mid)

WL = getSum(x[], w[], start, mid)

WR = getSum(x[], w[], mid+1, end)

if(WL < 0.5 && WR < 0.5):

return x[mid]

else if(WL > 0.5):

w[mid] += WR

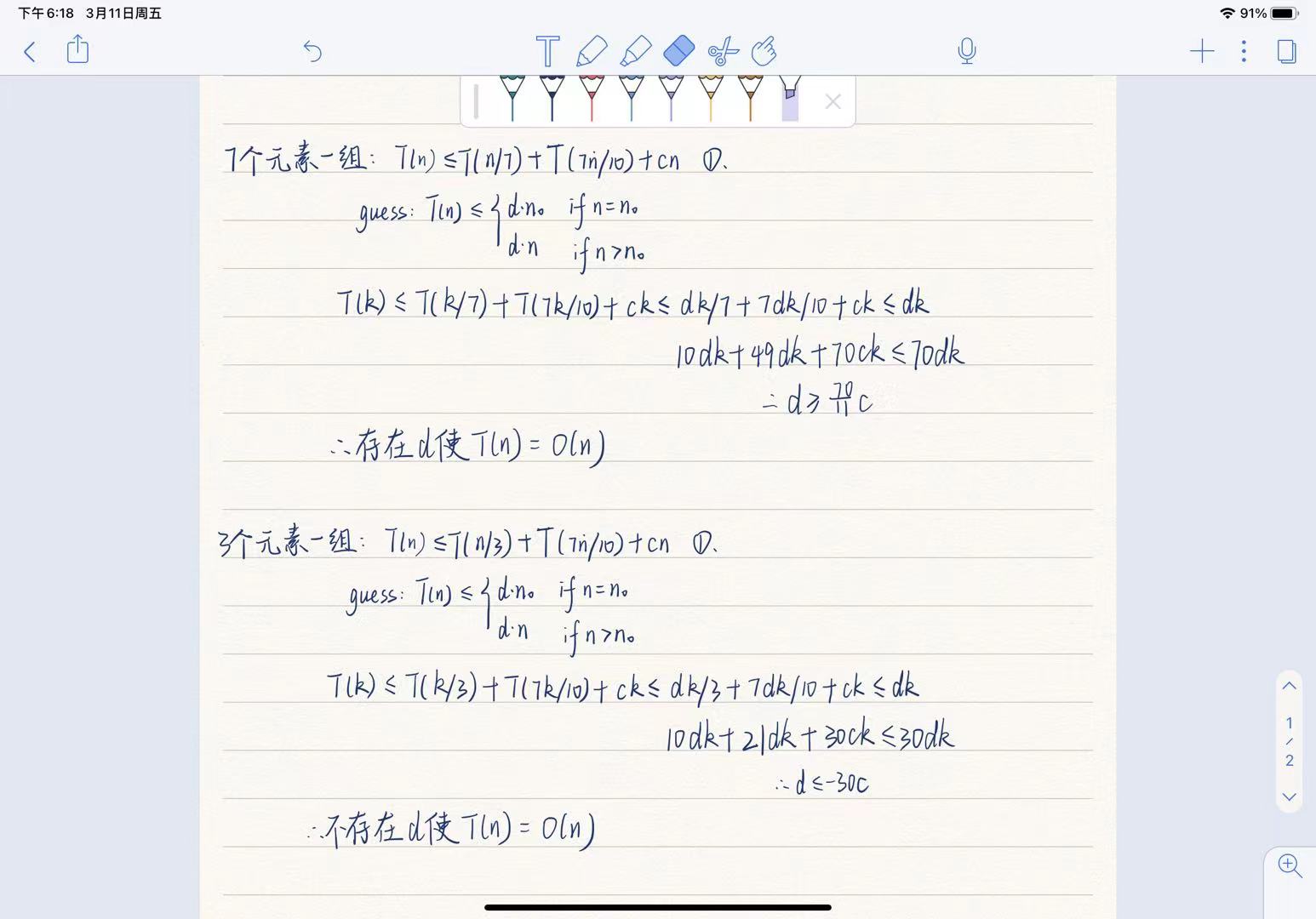
return WeightedMedian(x[], w[], start, mid)

else:

w[mid] += WL

return WeightedMedian(x[], w[], mid, end)

6.4 七个元素一组时，仿照五个元素一组时的证明方法，发现仍然是O(n)时间运行，但是性能可能不如五个元素一组好。



但是三个元素一组时，用替代法可以发现左边是大于d\*n的，所以不再是线性时间了。

