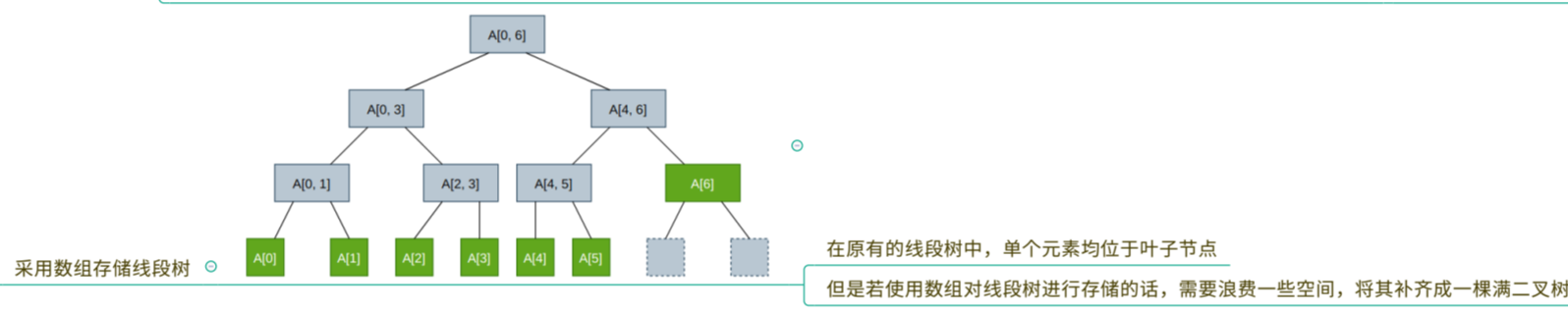
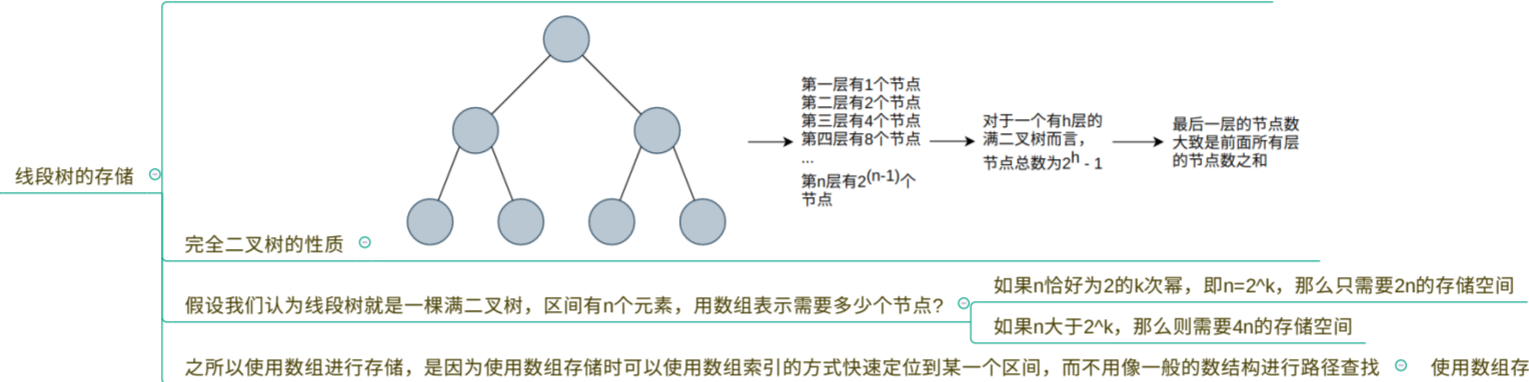
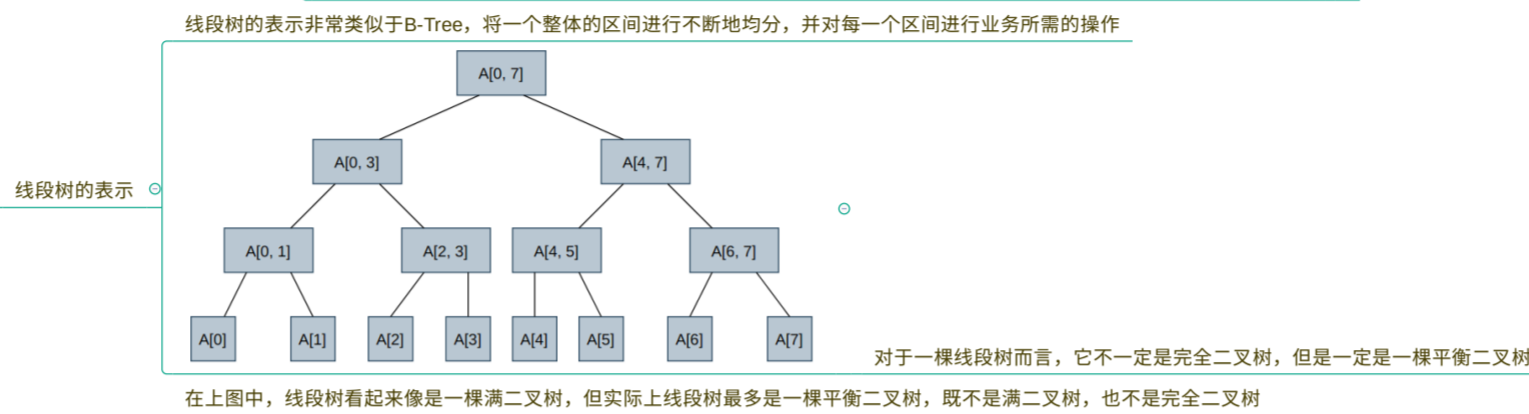
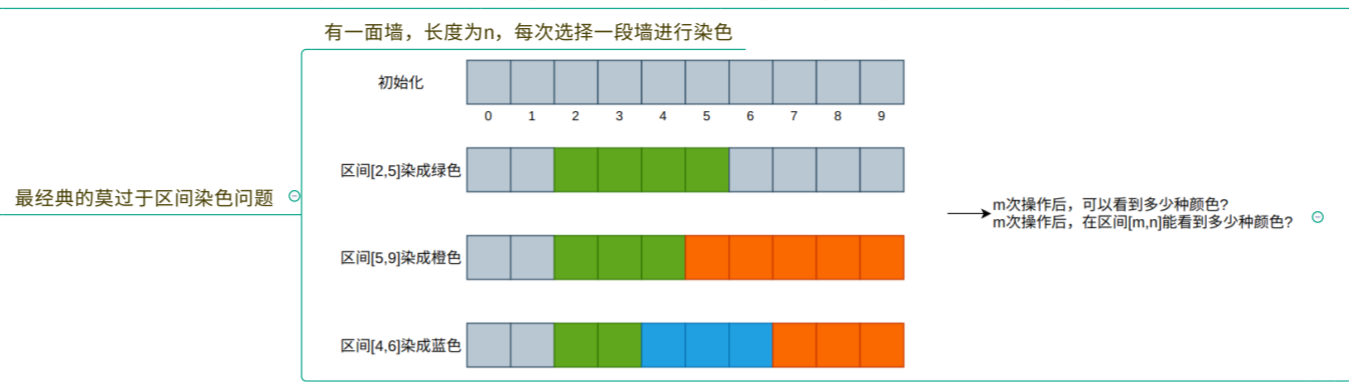


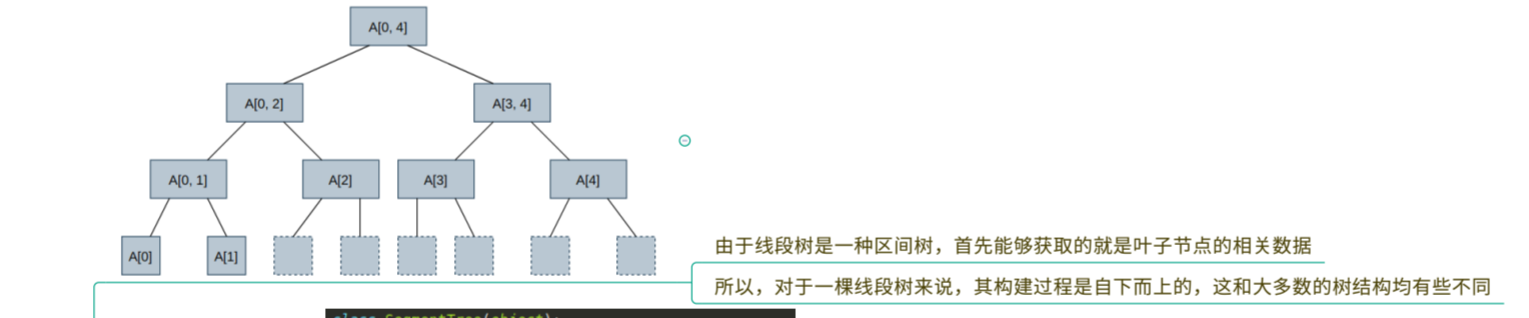
线段树(Segment Tree)

基本概念

线段树也称之为区间树，其目的在于解决区间问题。罗志祥多人运动问题。已知罗志祥每晚要约好几个女生到酒店房间。每个女生与罗志祥越好的时间由[start_i, end_i]表示，其中start_i表示女生进入房间的时间，end_i表示女生离开房间的时间。由于罗志祥同志心胸开阔，思想开明，不同女生可以同时存在于罗志祥的房间。请计算出罗志祥最多同时在做几人的多人运动



线段树的相关操作



```

class SegmentTree(object):
    def __init__(self, values):
        self.data = values
        self.tree = [None] * (len(values) * 4)
        self._build(0, 0, len(values) - 1)

    def _build(self, tree_index, left, right):
        # 递归终止条件
        if left == right:
            self.tree[tree_index] = self.data[left]
            return

        left_child = self._left_child(tree_index)
        right_child = self._right_child(tree_index)

        middle = left + ((right - left) >> 1)
        self._build(left_child, left, middle)
        self._build(right_child, middle + 1, right)

        # 递归返回并处理节点的值
        self.tree[tree_index] = self.tree[left_child] + self.tree[right_child]

    def interval_query(self, query_left, query_right):
        # query_left, query_right 均包含在区间内
        return self._query(0, len(self.data) - 1, query_left, query_right)

    def _query(self, tree_index, left, right, query_left, query_right):
        # 在以tree_index为根的线段树中[left, right]的范围内
        # 查询[query_left, query_right]区间的值

        if left == query_left and right == query_right:
            return self.tree[tree_index]

        left_child = self._left_child(tree_index)
        right_child = self._right_child(tree_index)
        middle = left + ((right - left) >> 1)

        if query_left >= middle + 1:
            return self._query(right_child, middle + 1, right, query_left, query_right)
        elif query_right <= middle:
            return self._query(left_child, left, middle, query_left, query_right)

        left_result = self._query(left_child, left, middle, query_left, middle)
        right_result = self._query(right_child, middle + 1, right, middle + 1, query_right)

        # 合并左右子树的结果
        return left_result + right_result
    
```