

分析：

①首次适应算法：FF算法要求空闲分区链以地址递增的次序链接。在分配内存时，从链首开始顺序查找，直到找到一个大小能满足要求的空闲分区为止。然后按照作业的大小，从该分区中划出一块内存空间，分配给作业，余下的空闲区仍留在空闲链中。若从链首到链尾都不能找到一个能满足要求的分区，则表明系统中没有足够大的内存分配给作业，内存分配失败，返回。

故该题目从链首20（又为表头节点，是第一个空闲分区30的分区起址）开始顺序查找，按空闲区30，20，5，46的顺序划出空间分配给作业A，B，C。

②最佳适应算法：BF算法要求将所有的空闲分区按期容量以从小到大的顺序形成一空闲分区链。每次为作业分配内存时，总能把能满足要求、又是最小的空闲分配给作业。

故该题目中160作为最小分区5的起址为链首（表头节点），将空闲区按从小到大的顺序排列5，20，30，46，然后分配给作业A,B,C。

③最坏适应算法：它在扫描整个空闲分区或链表时，总是挑选一个最大的空闲区，从中分割一部分存储空间给作业使用，以至于存储器中缺乏大的空闲分区。

故该题目中46是最大的空闲区，它的分区起址为210，所以由210作为链首（表头节点），将余下的空闲区从大到小排列，46，30，20，5，分配给作业A,B,C。