Algorithm Anaysis:

1.) Algorithm makeFirst(Position p)

```
DNode pos = checkPosition(p);

DNode posPrev = pos.getPrev();

DNode posNext = pos.getNext()

posNext.setPrev(posPrev);

prev.setNext(posNext);

pos.setPrev(header);
pos.setNext(header.getNext());
header.getNext().setPrev(pos);
header.setNext(pos);
```

2.) Algorithm boolean hasK(Sequence S, int k)

```
If(S.isEmpty())
        Return false;

if(k = S.first().element())

        return true;

else{

remove(S.first())

return hasK(S, k)
```

3.) If every element in the list is accessed every time the list is accessed, then all of the elements will have been accessed k times. Thus, the minimum is 0.
For the maximum number of elements accessed less than k times, if only one element is accessed every time the list is accessed, then all but one element will have been accessed less than k times. Thus, the maximum is n – 1.