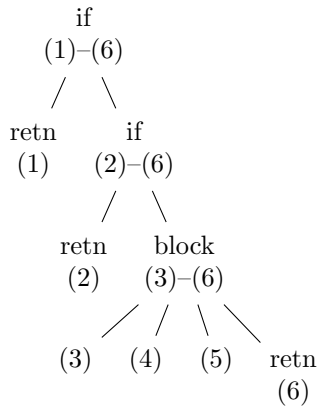
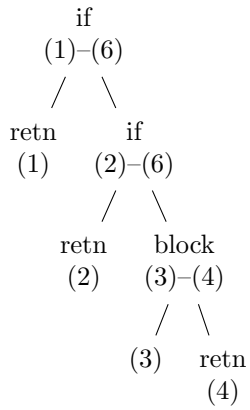
1.

a) Lines (3) through (6) take $O(1) + T(n-1)$ time. Lines (2) through (6) and (1) through (6) do as well.



b) Lines (3) through (4) take $O(n) + 2T(n/2)$ time. Lines (2) through (4) and (1) through (4) do as well.



2.

```
LIST split(LIST list, int n)
{
    LIST rest;

    if (n == 0) {
        rest = list->next;
        list->next = NULL;
        return rest;
    }
    else return split(list->next, n-1);
}

LIST kmergesort(LIST list, int k)
{
```

```
    if (k < 2) return NULL;
    else if (list == NULL) return NULL;
    else if (list->next == NULL) return list;
    else return kmerge(list, length(list), k, k-1);
}

LIST kmerge(LIST list, int len, int k, int n)
{
    LIST SecondList;

    if (list == NULL) return NULL;
    else if (list->next == NULL) return list;
    else {
        SecondList = split(list, n*len/k);
        return merge(kmergesort(SecondList, k),
                     kmerge(list, len, k, --n));
    }
}
```