(a) A naïve programmer writes the following Prolog program to implement a quicksort.

```
quicksort([], []).
quicksort([X|Tail], Sorted) :-
    split(X, Tail, Small, Big),
    append(SortedSmall, [X|SortedBig], Sorted),
    quicksort(Small, SortedSmall),
    quicksort(Big, SortedBig).

split( X, [], [], [X]).

split( X, [Y|Tail], [Y|Small], Big) :-
    X>Y, !,
    split(X, Tail, Small, Big).

split( X, [Y|Tail], Small, [Y|Big]) :-
    split(X, Tail, Small, Big).
```

Unfortunately, there are two mistakes that will prevent it running as expected. What are these mistakes and how can they be corrected? [6 marks]

(b) Explain how the operator ! in the split predicate works and why it is used here. [2 marks]

(c) Our programmer now decides to improve the efficiency of the program by using difference lists. Explain how the technique works and modify the program to use difference lists by introducing a new predicate quicksort2.

```
quicksort( List, Sorted) :- quicksort2( List, Sorted - []).
```

[6 marks]

(d) Comment on the space and time complexity of the execution of the two versions of quicksort for the call quicksort([2,5,7], X). [6 marks]