1998 Paper 1 Question 6

Foundations of Computer Science

This question has been translated from Standard ML to OCaml

What does $O(g(n))$ mean, and what is its relevance to programming? (Describe both advantages and limitations.) [5 marks]

Consider the following OCaml declarations, for tree-like expressions:

```
type 'a expr = Join of 'a expr * 'a expr
             | Tip  of 'a

let rec flatten = function
  | Tip x         -> [x]
  | Join (e1, e2) -> flatten e1 @ flatten e2
```

The *size* of an expression is the number of `Tip`s it contains. State the complexity of `flatten` e, measured in cons operations, as a function of the size of e:

(a) in the worst case [3 marks]

(b) in the average case [4 marks]

(c) in the best case [3 marks]

Code a function `flat` such that `flat` $e =$ `flatten` e for all e, justifying this claim. Show that `flat`'s worst-case complexity is linear. [5 marks]