7. [5 points] Assume the following type declaration:

```
data Tree a = Tip | Bin (Tree a) a (Tree a)
```

Write a function `fringe :: Tree a -> [a]` that returns the values of the exterior node
(those with 0 or 1 children), in left-to-right order. Thus,

```
fringe Tip = []
fringe (Bin (Bin Tip 1 Tip) 2 (Bin Tip 3 (Bin Tip 4 Tip))) = [1,3,4
```

10. [3 points] Assume the following Prolog code:

```
legs(cat, 4).
legs(elephant, 4).
legs(bear, 2).
legs(bear, 4).
legs(hawk, 2).
legs(moth, 6).
legs(snake, 0).

flies(hawk).
flies(moth).

climbs(cat).
climbs(bear).
climbs(snake).

high_places(A) :- flies(A).
high_places(A) :- climbs(A).
```

If we ask Prolog for all solutions to `high_places(X), legs(X, L), dif(L, 4).` what values will occur for X?

not equal