tgao

---

as lecture says haskell we can define recursively to find number divisible by 6, so I can operate list from the end to front or front to end, every time I only care about first / last element in list and determine whether it's divisible by 6 or not, if yes, count it, if no, recurisive the list start from the next/pervious element.at the end you can get the total number in this list which divvisible by 6 , you have to define the input type and output type correct.

---

you don't have to worry about the entire list, only care about portion on list(eg. head, tail) at a time. once you meet the condition you can jump out quicker

---

as lecture mentioned haskell use lazy evaluation, it only evaluate when it's needed, by doing this it saves time and space without doing extra work, it will cost a longer time if you evaluate all the functions beforehand.