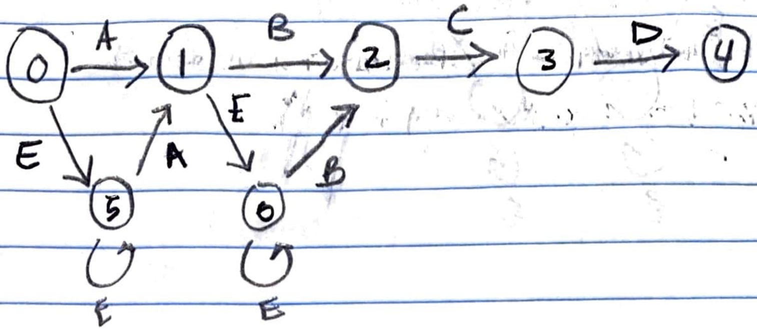
Module 5 10/10/22

Exercise 5.3 : We can skip 4 letters

Exercise 5.4



Exercise 5.7

```
Algorithm : Match (String a, String b)
1. IF (a.length() = 0 & b.length() = 0)
      return true
   IF (a.length >1 & b.charAt (0) = '*' ) {
      int i = 0
      while (i+1 ≤ a.length() & a.
         i++
      a = a.substring(i)
   }
   IF (a.length() > 1 & a.charAt (0) = '*' & b.length() = 0)
      return false
   if (a.length() > 0 & a.charAt. '*' )
      return match (a.substring (1), b) ||
             match (a, b.substring (1))
```

Exercise 5.5 D A (B|C) E*

Exercise 5.1

```
for loop version : complexity  O(n · m)
  findFirstOccurance (char[] pattern, char[] text)
   for i from 0 to text. length
    if text[i] = pattern [0]
     int; int num = i; boolean match = true
     for j=0 to patter. length {
        if text [i+j] != patter[j]
           Match = false
     }
       if Match = true
         return true
    }
   }
   return -1
```

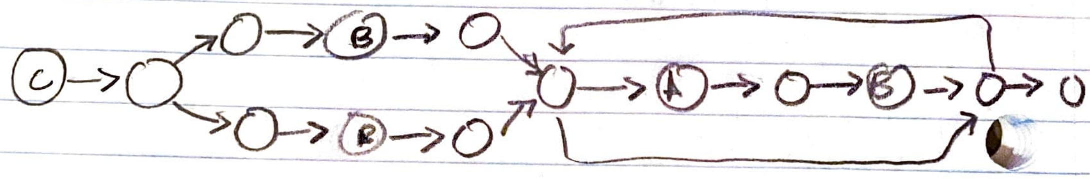
```
While loop version : complexity  O(n)
  find FirstOccurance (char[] p , char[] t)
   Int i = 0, Int c = 0, Int num = 0
   While i != t. length {
     If p[c] = t.[i] {
       If c = 0
         num = i
       i++
     }
      else
        c = 0
       i++
    }
    If
      return c = t. length-1
    else
      return -1
```

recursive version : Complexity O(n)

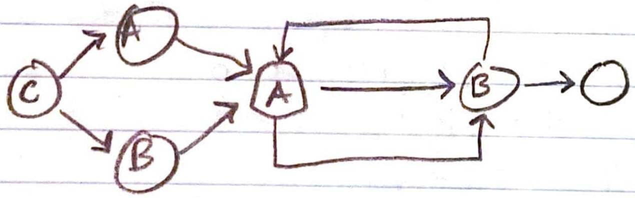
```
findFirstOccurance ( char[] p, char [] t)
    If p.length = 0
        return true
    If t[0] = p[0]
        findFirstOccurance ( t. copyOfRange (t,1,t. length-1),
                             p. copyOfRange (p, 1, p.length -1)
```

Exercise 5.6

Unoptomized:



Optomized



Exercise 5.8

The page rank algorithm works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The main disadvantage is that it favors older pages because a new page may not have many links