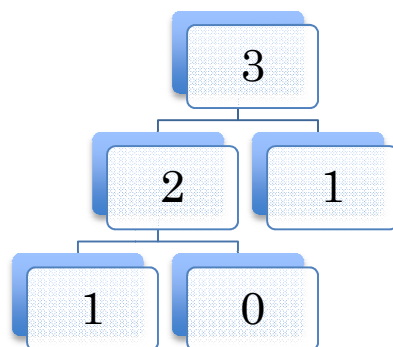


問題 1

(a)



(b)

```
def F(n)
def F(n)
  if n > 1
    F(n-1) + F(n-2) + 2
  else
    0
  end
end
end
```

(c)fib(n)

(d) $F(n) = f(n) * 2 - 2$

理由：上記の木構造に対して、 $n > 1$ のとき下に二つのノードと二つのエッジが生成される(☆)。このエッジの総数が $F(n)$ であり、生成された最下層のノードの総数が $f(n)$ である。今一つの根から $f(n)$ 個の最下層のノードを生成するには、 $f(n)-1$ 回の(☆)が行われなければならない。そのため生成されるエッジの総数は $F(n) = (f(n) - 1) * 2$

解説

木構造を書けば一目瞭然だけど、 $f(0)$ or $f(1)$ の 1 をかき集めているので、 $fib(n)$ の葉は $fib(n)$ 個です。

問題 2

(a)(ア) $a[i][j] - a[k][j] * aik$ (イ) $a[i][col-1]$

(b) akk の値に 0 が入ると数値を 0 で割ることになってしまい正しい解が得られない。また 0 に近い数字が入ってしまうと割り算の結果が非常に大きくなってしまい、大きな数同士の引き算の結果情報落ち誤差が生じてしまう。 akk にそのような値が極力入らないように $|a[k][k]|$ の大きな行から扱うようにすればよい。

解説

線形代数でやる掃き出し法です。複雑に書いてあるけどやることは同じです。教科書の通り。(見落としてたので修正)

問題 3

(a) ア 3 イ 1 ウ -1 エ 1 オ 2 カ 3 キ 2 ク 0 ケ 4

(b)

AGTGGGA
AG—GTAA

解説

頑張れ。7章を参照の事。

簡単に説明すると、一つ上-2,一つ左-2,一つ左上から(一致なら+2,不一致なら-1)して表を埋めていき、最後に左下からたどればいいです。

問題 4

(a)(ア)オブジェクト内部のデータを隠蔽したり、オブジェクトの振る舞いを隠蔽したり、オブジェクトの実際の型を隠蔽したりすること。

(イ)データの変更の影響を局所にとどめ、また概念を抽象化するため。

(b)(ア)あるオブジェクトが他のオブジェクトの特性を引き継ぐこと。

(イ)一度定義した属性や機能を再び宣言することなく利用するため。

(c)(ア)複数の型に属し、統一的に扱うことが可能であること。

(イ)同じメッセージに対して、そのメッセージの受け手によって適切なメソッドを選び出すため。

解説

教科書にはっきりとした説明が見当たりません。そのため主に [wikipedia](#) より引用しています。(つまり内容が絶対だと保証はできません。多分あってるけど。)でもなんとなく分かってくれると嬉しいな。オブジェクト指向の利点をあげていけばいいでしょう。