1. (a)

   BAR/BAR/BAR = $\frac{1}{4} \times \frac{1}{4} \times \frac{1}{4} \times 20 = \frac{20}{64}$

   BELL/BELL/BELL = $\frac{1}{4} \times \frac{1}{4} \times \frac{1}{4} \times 15 = \frac{15}{64}$

   LEMON/LEMON/LEMON = $\frac{1}{4} \times \frac{1}{4} \times \frac{1}{4} \times 5 = \frac{5}{64}$

   CHERRY/CHERRY/CHERRY = $\frac{1}{4} \times \frac{1}{4} \times \frac{1}{4} \times 3 = \frac{3}{64}$

   CHERRY/CHERRY/? = $\frac{1}{4} \times \frac{1}{4} \times \frac{3}{4} \times 2 = \frac{6}{64}$

   CHERRY/?/? = $(\frac{1}{4} \times \frac{3}{4} \times \frac{3}{4} + \frac{1}{4} \times \frac{3}{4} \times \frac{1}{4}) \times 1 = \frac{12}{64}$

   期望值：

   $$\frac{20+15+5+3+6+12}{64} = \frac{61}{64} \tag{1}$$

   (b)

   BAR/BAR/BAR: $\frac{1}{4} \times \frac{1}{4} \times \frac{1}{4} = \frac{1}{64}$

   BELL/BELL/BELL: $\frac{1}{4} \times \frac{1}{4} \times \frac{1}{4} = \frac{1}{64}$

   LEMON/LEMON/LEMON: $\frac{1}{4} \times \frac{1}{4} \times \frac{1}{4} = \frac{1}{64}$

   CHERRY/CHERRY/CHERRY: $\frac{1}{4} \times \frac{1}{4} \times \frac{1}{4} = \frac{1}{64}$

   CHERRY/CHERRY/? = $\frac{1}{4} \times \frac{1}{4} \times \frac{3}{4} = \frac{3}{64}$

   CHERRY/?/? = $\frac{1}{4} \times \frac{3}{4} \times \frac{3}{4} + \frac{1}{4} \times \frac{3}{4} \times \frac{1}{4} = \frac{12}{64}$

   總合 =

   $$\frac{1+1+1+1+3+12}{64} = \frac{19}{64} \tag{2}$$

   (c) mean = 216.994500 median = 21 (大約啦，median比較固定 mean 會比較飄)

```go
package main

import (
    "fmt"
    "math/rand"
    "sort"
    "time"
)

const (
    BAR    = iota
    BELL   = iota
    LEMON  = iota
    CHERRY = iota
)

func playOnce() int {
    remainder, plays := 10, 0
    choices := []int{BAR, BELL, LEMON, CHERRY}
    slots := [3]int{}
    var equalType int

    for remainder > 0 {
```

```
        remainder -= 1
        plays += 1
        for i := 0; i < 3; i++ {
            slots[i] = choices[rand.Intn(len(choices))]
        }

        // evaluate
        if slots[0] == slots[1] && slots[1] == slots[2] {
            equalType = 3
        } else if slots[0] == slots[1] {
            equalType = 2
        } else {
            equalType = 1
        }

        if slots[0] == CHERRY {
            remainder += equalType
        } else if equalType == 3 {
            if slots[0] == BAR {
                remainder += 20
            } else if slots[0] == BELL {
                remainder += 15
            } else {
                remainder += 5
            }
        }
    }
    return plays
}

func play(times int) {
    results := []int{}
    sum := 0
    for i := 0; i < times; i++ {
        res := playOnce()
        results = append(results, res)
        sum += res
    }
    sort.Ints(results)
    mean := float64(sum) / float64(len(results))
    median := results[len(results)/2]
    fmt.Printf("mean = %f median = %d\n", mean, median)
}

func main() {
    rand.Seed(time.Now().UnixNano())
    play(50000)
}
```

2.

	陽性	陰性
染病	0.99	0.01
沒有染病	0.01	0.99

p(染病|陽性) = p(染病, 陽性) / p(陽性) = 0.0001× 0.99 / (0.9999 × 0.01 + 0.0001 × 0.99) = 0.009804

好消息是大約只有 1% 不到的機率是真的染病

3. (a)
   從數值上可以計算出 p(Burglary, Earthquake) = p(Burglary) p(earth)。從拓撲上可以看出 Burglary 和 Earthquake 沒有直接的連線。
   (b)
   只要檢查 p(b,e|a) 是不是等於 p(b|a)p(e|a) 就可以了

情況	值
$p(b,e\|a)=\alpha p(a,b,e)=\alpha p(a\|b,e)p(b,e)$	$p(b,e\|a)=0.95\times0.001\times0.002\alpha=0.0008\alpha$
$p(b,\neg e\|a)=\alpha p(a,b,\neg e)=\alpha p(a\|b,\neg e)p(b,\neg e)$	$p(b,\neg e\|a)=0.94\times0.001\times0.998\alpha=0.3728\alpha$
$p(\neg b,e\|a)=\alpha p(\neg b,e,a)=\alpha p(a\|\neg b,e)p(\neg b,e)$	$p(\neg b,e\|a)=0.29\times0.999\times0.002\alpha=0.2303\alpha$
$p(\neg b,\neg e\|a)=\alpha p(\neg b,\neg e,a)=\alpha p(a\|\neg b,\neg e)p(\neg b,\neg e)$	$p(\neg b,\neg e\|a)=0.001\times0.999\times0.998\alpha=0.3962\alpha$

$$p(b,e|a)=0.0008\alpha \quad (3)$$

$$p(b|a)=p(b,e|a)+p(b,\neg e|a)=(0.0008+0.3728)\alpha=0.3736\alpha \quad (4)$$

$$p(e|a)=p(b,e|a)+p(\neg b,e|a)=(0.0008+0.2303)\alpha=0.2311\alpha \quad (5)$$

$$p(b,e|a)=0.0008\neq0.0863=0.3736\times0.2311=p(b|a)p(e|a) \quad (6)$$

Burglary and Earthquake are not independent.

4. (a) (ii) (iii)
   (b)

$$p(b,i,\neg m,g,j)=p(b)p(\neg m)p(i|b,\neg m)p(g|b,i,\neg m)p(j|g) \quad (7)$$

$$p(b,i,\neg m,g,j)=0.9\times0.9\times0.5\times0.8\times0.9=0.2916 \quad (8)$$

   (c)

$$p(J|b,i,m)=\alpha\big(p(J,g)+p(J,\neg g)\big) \quad (9)$$

$$p(j|b,i,m)=\alpha(0.81+0)=0.81\alpha \quad (10)$$

$$p(\neg j|b,i,m)=\alpha(0.09+0.1)=0.19\alpha \quad (11)$$

   The probability of going to jail is 0.81

   (d) 沒有罪或沒有被起訴不用赦免，所以應是把 P 加在 G 和 I 之後再把 P 連到 J

5. (a) (ii) (iii)
   (b). (iii)
   (c) Yes. Because Wrapper and Shape are d-separated (indpendent)
   (d) strawberry: $70\%\times80\%=56\%$
   anchovy: $30\%\times10\%=3\%$
   sum = 59%
   (e) p(strawberry|red, round) = p(straberry, red, round)/p(red,round)

$$(70\%\times80\%\times80\%)\div(70\%\times80\%\times80\%+30\%\times10\%\times10\%)=448\div451\approx0.993 \quad (12)$$

   (f) $0.7s+0.3a$

   (g) The value is same