Problem 1

b) Example where there is no stable pairs

Network A has TV shows $(5, 15)$
Network B has TV shows $(10, 20)$

Only two possible schedules:

1^{st}: $(5, 10)$ and $(15, 20)$ where Network B wins both time slots
Then, A could swap time slots of their shows, ending up with $(5, 15)$ and $(10, 20)$ where both networks win one slot.

2^{nd}: $(5, 20)$ and $(10, 15)$ where both win one slot. Network B can now swap time slots of their shows getting $(5, 10)$ and $(15, 20)$ and they win both.

It is shown that both these scenarios one network can unilaterally change their schedule and win more time slots.

Problem 2

- While h_i has position available
  - h_i offers position to next student S_i on prefrence list
    - If S_i is available
      - S_i accepts
    - Else S_i already committed to hospital h_k
      - If h_k is higher prefrence than h_i
        - S_i remains in h_k
      - Else
        - S_i accepts h_i offer
        - Positions in h_k go up 1
        - Positions in h_i go down 1

- This follows very closely to the GS algorithm. I am performing a matching between current hospital and all of its students.

Problem 3

What is known

- No two ships can be in same port
- days in month greater than ships
- Truncated port cant be used again

• Each port must have priority list for each ship

For port P_i in all ports
  P_i offers spot to ship S_j highest in its priority list
  if S_j has not been matched
    S_j accepts
  else S_j is already matched with P_k
    if P_i is higher in S_j's schedule
      S_j commits to P_i
      P_k now has availability for accepting another ship
    else P_k is higher than P_i in S_j's schedule
      S_j remains committed to P_k

• This is stable, if there was an unstable matching: S_i passes P_k after S_j has already stopped at this port. As P_k is higher in S_i's priority list, and vice-versa. Proven by contradiction.

Problem 4 (There may have been typo in book in labeling the functions)

Here are the functions ordered in ascending order

- $g_1(n) = 2^{\sqrt{\log n}}$
- $g_3(n) = n(\log n)^3$
- $g_4(n) = n^{4/3}$
- $g_5(n) = n^{\log n}$
- $g_2(n) = 2^n$
- $g_7(n) = 2^{n^2}$
- $g_6(n) = 2^{2^n}$

Reasoning:

- Easy ones to tell were polynomically-bounded and exponental functions

$$n(\log n)^3 = O(n^{4/3}) \quad 2^n = O(2^{n^2}) \quad 2^{n^2} = O(2^{2^n})$$

- Assuming logarithms are base 2,

$$2^{\sqrt{\log n}} = O(2^{\log n}),\ 2^{\log n} = n \text{ so } 2^{\sqrt{\log n}} \text{ is sub linear}$$

- $n^{4/3} = O(n^{\log n})$ since $4/3 = O(\log n)$
- $n^{\log n} = 2^{\log(n^{\log n})} = 2^{(\log n)(\log n)} = 2^{(\log n)^2} = O(2^n)$

Problem 5

a) $1+2+\ldots+n = \frac{n(n+1)}{2}$

Base: $n=1$ $\sum_{i=1}^{n} i = \frac{1(1+1)}{2}$

$1 = \frac{2}{2}$ $1=1$ ✓

Inductive: Assume $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ for $n \geq 1$

$$\sum_{i=1}^{n+1} i = \sum_{i=1}^{n} i + (n+1) = \frac{(n+1)(n+2)}{2}$$

$$= \frac{n(n+1)}{2} + (n+1) = \frac{(n+1)(n+2)}{2}$$

$$\frac{n^2+n}{2} + \frac{2(n+1)}{2} = \frac{n^2+3n+2}{2}$$

$$\frac{n^2+3n+2}{2} = \frac{n^2+3n+2}{2}$$ ✓ Proven by induction

b. $1^3+2^3+3^3 \ldots n^3 = ??$

$n=1 \longrightarrow 1 \; (1)^2$

$n=2 \longrightarrow 9 \; (1+2)^2$

$n=3 \longrightarrow 64 \; (1+2+3)^2$

$n=4 \longrightarrow 100 \; (1+2+3+4)^2$

$n=5 \longrightarrow 225 \; (1+2+3+4+5)^2$

$$\sum_{i=1}^{n} i^3 = \left(\sum_{i=1}^{n} i\right)^2$$

$$= \left(\frac{n(n+1)}{2}\right)^2$$

$$\boxed{\sum_{i=1}^{n} i^3 = \frac{n^2(n+1)^2}{4}}$$ now proving this

> Base: $n=1$ $\sum_{i=1}^{n} i^3 = \frac{1(1+1)^2}{4}$

$1 = \frac{4}{4}$ $1=1$ ✓

Inductive: Assume: $\sum_{i=1}^{n} i^3 = \frac{n^2(n+1)^2}{4}$

$$\sum_{i=1}^{n+1} i^3 = \sum_{i=1}^{n} i^3 + (n+1)^3 = \frac{(n+1)^2(n+2)^2}{4}$$

$$= \frac{n^2(n+1)^2}{4} + (n+1)^3 = \frac{(n+1)^2(n+1)^2}{4}$$

$$= \frac{n^2(n+1)^2}{4} + (n+1)^3 = \frac{(n+1)^2(n^2+4n+4)}{4}$$

$$= \frac{n^2(n+1)^2+4(n+1)^3}{4} = \frac{(n+1)^2(n^2+4n+4)}{4}$$

$$= \frac{(n+1)^2(n^2+4n+4)}{4} = \frac{(n+1)^2(n^2+4n+4)}{4}$$ ✓ The sum is proven by induction

Problem 6

a. 200 steps

You would need 20 tries for worst case.

	1st egg	If 1st egg breaks → 2nd egg	Drops
	20	1→2→3→4→…→19	1+19=20
19 C	39	21→22→23…→38	2+18=20
18 C	57	40→41→…→56	3+17=20
17 C	74	58→60→…→74	4+16=20
	90		20
	105		
	119		
	132		
	144		
	155		
	165		
	174		
	182		
	189		
	195	190→191→…→194	15+5=20
	200	196→197→198→199	16+4=20

b. n steps

For n steps you take an inital step value x, and make the subsequent step be $x-1$.

$$x + x-1 + x-2 \ldots + 1 = n$$

This is $\frac{x(x+1)}{2} = n$

x must be an integer and solving for x gives you $x = \frac{-1+\sqrt{1+8n}}{2}$

plugging in n value gets you how many tries it will take.