① For every set of TV shows and ratings is there always a stable pair of schedule?

Sun Yi [illegible]
(UID: 5051467[illegible])

Pf

I will give an example of a set of TV shows and associated ratings to show that there is no stable pair of schedules.

Assume Network A and Network B each has 2 shows.
Network A has $\{x_1, x_2\}$ for the two shows it owns.
Network B has $\{y_1, y_2\}$ for the two shows it owns
The ratings for the shows of each network is shown below

The order of the shows does not matter and it is for display purpose only.

Shows	Ratings
x_1	60
x_2	80
y_1	40
y_2	70

60, 40
80, 70

60, 70
80, 40

For this example, either Network A or B will reveal one of the two scheduled shows. If the result pair is show x_1 versus y_1 and show x_2 versus y_2. Network B will definitely want to switch the order of its shows in order to win one time slot, instead of none time slots. If the result pair is show x_1 versus y_2 and show x_2 versus y_1. Network A will definitely want to switch the order of its shows in order to win both time slots instead of win one time slot.

② This is the algorithm (Based from the Gale Shapley algorithm)
Throughout the entire algorithm, a student could be taken the offer from a hospital or remain available. Similarly, a hospital could still be in the hiring process or all the available positions are filled.

Initially all students are free and all hospital haven't started the hiring process

while some arbitrary hospital h_a is hiring to fill available positions.
    h_a will offer the next highest prefered student S_b a position from its list
    if student S_b is available to accept the position
        S_b will take the offer and accept the position
    else S_b has already accepted a position from another hospital h_c
        if S_b prefers h_c over h_a according to the preference list
            S_b will keep the position offer from hospital h_c
        else S_b prefers h_a over h_c according to the preference list.
            We will increase the number of free positions at h_c by one
            We will decrease the number of free positions at h_a by one

---

Analysis: The worst case of the algorithm is $O(mn)$.
Each hospital gives out a position offer to at most one student
Each time the algorithm runs, some arbitrary hospital offers a position to some arbitrary student. Let's assume that there are P_a available positions at hospital h_a (where $P_a > 0$). The algorithm will finish when all the hospital's positions are taken because any hospital will keep offer the next highest preferred students a position if the position is still available. But throughout the process, an arbitrary student could have committed to another hospital h_b. Therefore, the worst case is that a hospital need to go through n students in order to fill the positions. $\therefore$ steps $= O(m\,n)$

We know that the total number of positions are less than n students

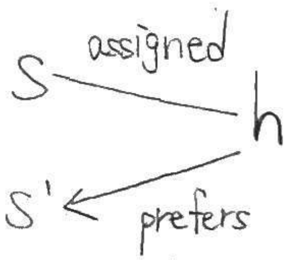
m: total number of hospitals
n: number of students

(2) cont.

Show the assignment is stable

Case 1

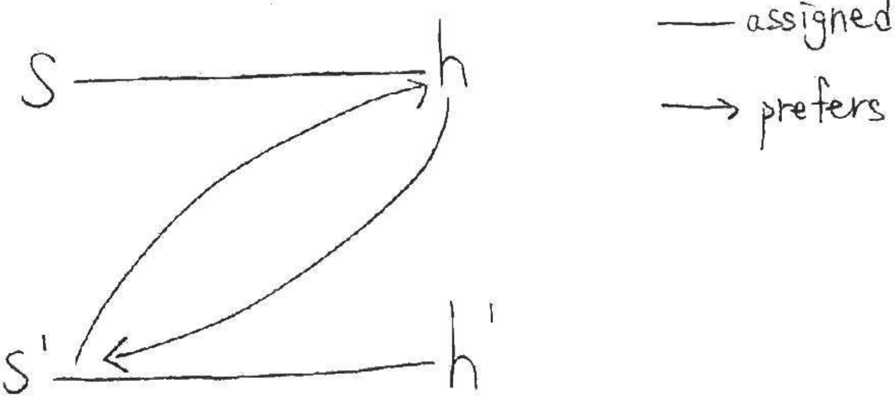
Proof by contradiction. Assume the assignment is unstable. According to first kind of instability, given student s and s' and a hospital h. A diagram is shown below



According to the preference list, if hospital h prefers student s' to s, then it means that hospital h would has sent out an offer to student s' before student s. Therefore, student s' becomes unavailble (not free) after accepting the offer from some hospital. However, this contradict our instability situation that student s' should have been free at the end.

Case 2

Proof by contradiction. Assume the assignment is unstable. According to the second type of instability, given any arbitrary pair of student s & s' and hospital h & h'. A diagram is shown below.



next page →

Assume that there is an arbitrary pair (h_a, S_b) that is instable. Since there are some other hospitals also prefers student S_b, it means hospital h_a must have given an offer to student S_b. However, student S_b reject h_a for some arbitrary h_c according to the preference list. At the end, student S_b is settled with the position offered by some other arbitrary h_d (given $h_d \neq h_c$) Therefore, student S_b prefers h_d over h_a.

$$h_c > h_a$$

$$h_d > h_a \rightarrow (h_c \neq h_d)$$

$$h_d > h_a$$

This contradict our assumption that the hospital h_d that student S_b committed to ranks higher than hospital h_a on the preference list. ■

③ The algorithm is define as follows:
For any arbitrary ship schedule, we need to decide a maintenance port which is remains at the port for the rest of the month. The truncations schedule of each ship depend on the assigned maintenance port. The assignment of the maintenance port of each ship is <u>legit</u> (a proper assignment) when the rest of the truncations schedules meet the safety requirement. The safety requirement is "No two ships can be in the same port on the same day" This means there is no overlap in terms of the maintence port for each ship & each ship has their own unique maintenance port under legit assignment of the maintenance port. The algorithm is similar to the gale shapley matching algorithm. Each ship is gonna ranks each port on a preference list. The preference list from each ship is based on the time sequence in which the ship visit each port.
Each port is gonna rank each ship on a preference list. The preference list from each port is based on the reverse time sequence in which the ship visit the port.

Show a set of truncations can always be found which means showing: the definition of a legit assignment of the maintenance port is a stable matching between any arbitrary pair of ship & port. (a stable matching assume it is)

Pf by contradiction:
Assume the assignment of the maintenance port is not legit which means it does not meet the safety requirement which is allowing two ships visit at the same port.
Some arbitrary ship S_a has stopped at port P_x and some arbitrary ship S_b visit port P_x on the same day.
According to our indicated preference rules, ship S_b prefers port P_x over its assigned maintenance port P_y. At the same time,

→

port P_x prefers ship S_b over its original assigned ship S_a. ② cont.
Therefore, $\{S_b, P_x\}$ is the instable pair.
This form a contradiction to our assumption that we have a stable matching.
∴ A stable matching between ships & ports requires a legit assignment of the maintenance port to each ship. ∎

There n ports & n ships.
A ship might need to pass through n-1 ports to find the one port to stay

$n(n-1) \approx n^2$

$\boxed{O(n^2)}$

(4) Arrange in ascending order of growth rate

$$\boxed{g_1, g_4, g_3, g_5, g_2, g_7, g_6}$$

(5a) Prove by induction that sum of the first n integers $(1+2+\dots+mn)$ is $n(n+1)/2$

① Check for base case, $n=1$

② Induction, show $n=k+1$ works by assuming $n=k$ works.

Let $S(n) = 1+2+\dots+mn = \frac{n(n+1)}{2}$

---

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

② Assume true for $n=k$

$$S(k) = \frac{k(k+1)}{2} = 1+2+3+\dots+k$$

Show it is true for $n=k+1$

$$S(k+1) = 1+2+3+\dots+k+k+1 = \frac{k(k+1)}{2} + k+1$$

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

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

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

∎

(56) Prove by induction that $1^3+2^3+3^3+\ldots+n^3=?$

Through trial & error, we know that $1^3+2^3+3^3+\ldots+n^3=\frac{n^2(n+1)^2}{4}$

Pf :

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

① Check for base case, $n=1$

$$s(1)=\frac{(1)^2(1+1)^2}{4}=\frac{(1)^2(1+1)^2}{4}=\frac{(1)(2)^2}{4}=\frac{\cancel{4}}{\cancel{4}}=1$$

② Assume true for $n=k$

$$s(k)=1^3+2^3+3^3+\ldots+k^3=\frac{k^2(k+1)^2}{4}$$

Show it is true for $n=k+1$

$$s(k+1)=1^3+2^3+3^3+\ldots+k^3+(k+1)^3=\frac{k^2(k+1)^2}{4}+(k+1)^3$$

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

$$=\frac{(k+1)^2[k^2+4(k+1)]}{4}$$

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

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

(6) The number of tries I need in the 2 egg problem when there are 200 steps :

Our goal is to reduce the worst case to attempt to make all cases take the same number of drops attemp to find out the solution.

Let assume we drop our first egg on y floors. If the egg breaks, then we can step through the rest of y-1 floors one at a time.

But if the egg doesn't break, we will move up y-1 floors because we will have one less drop attempt available and we do not want to go pass y number of attempts.

Therefore, the next floor we should try is y+(y-1)

Alike the first time, if the egg doesn't break, then the next floor we should try is y+(y-1). Similarly, the next floor we should try is y+(y-1)+(y-2), then floor y+(y-1)+(y-2)+(y-3)+....

We will reduce the step size by one each time we move up until the step size is one at the end. The following summation will give us the number tries I need in the 2 egg problem under 200 steps. :

$$y+(y-1)+(y-2)+(y-3)+\dots+1 \geq 200$$

Prove from problem 5a) $\longleftarrow$ $$\frac{y(y+1)}{2} \geq 200$$

$$y(y+1) \geq 400$$

$$y^2+y \geq 400$$

$$y^2+y-400 \geq 0$$

$$y = \frac{-1 \pm \sqrt{(1)^2-4(1)(-400)}}{2(1)}$$

$$y = \frac{-1 \pm \sqrt{1601}}{2} \qquad y = \frac{-1 \pm 40.012}{2}$$

$$y = 19.506 \approx \boxed{20 \text{ floors}}$$

For n steps, we gonna apply the same method from the previous page.
The number of tries I need in the 2 egg problem ~~under~~ k steps:

$$y+(y-1)+(y-2)+(y-3)+\dots+1 \geq k$$

Prove from problem 5a) $\leftarrow$ $\frac{y(y+1)}{2} \geq k$

$$y(y+1) \geq 2k$$

$$y^2+y \geq 2k$$

$$y^2+y-2k \geq 0$$

$$y=\frac{-1\pm\sqrt{(1)^2-4(1)(-2k)}}{2(1)}$$

$$y=\frac{-1\pm\sqrt{1+8k}}{2}$$

$$\boxed{y=\frac{-1+\sqrt{1+8k}}{2} \text{ floors}}$$