CS 180 HW 1

Wednesday, October 7, 2020 7:00 PM

Amy Seo

Dis. 1D
10:00-11:50 AM
Song

1. (B) Counterexample: Let $n=2$
   - Network A has 2 shows w/ ratings 5.0 and 3.0 respectively.
   - Network B has 2 shows w/ ratings 2.0 and 4.0 respectively.
   - Currently, each network has won one prime time slot.

set of TV shows

Network	A	B
Ratings	5.0	2.0
	3.0	4.0

- Network A can swap their 2 shows to win all 2 shows

A	B
3.0	2.0
5.0	4.0

- Network B can swap their 2 shows to win one slot

A	B
3.0	4.0
5.0	2.0

- Network A can swap again to win both slots

A	B
5.0	4.0
3.0	2.0

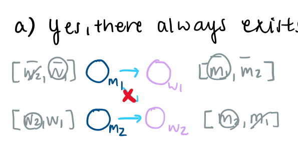
- Network B can swap again to win one slot

A	B
5.0	2.0
3.0	4.0

- We're back at square one, so this swapping will continue, making this set of TV shows instable.

∴ For every set of TV shows and ratings, there is not always a stable pair of schedules, as seen in the counterexample.

2. a) Yes, there always exists a perfect matching w/ no strong instability.



- While there is an unmarried man:
  - pick an arbitrary unmarried man m
  - while m is not married:
    - m proposes to the first woman w on his preference list that he has not yet proposed to
      → if w is not married
        - m and w are now married
      → else
        - if she is married and m is higher on her preference list than man m' whom she is currently married to, m and w are now married — CANNOT be tied!
        - m' is now single

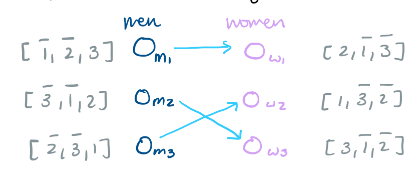
* Proof that this algorithm creates perfect matches:
1) A woman will always remain married once she gets her first proposal
2) A man will propose to women who are either tied or worse in preference to his previous partner(s)
3) Since there are equal amount of men and women, if a man is unmarried, it is guaranteed there is an unmarried woman to propose to.
   - if the unmarried man has proposed to every woman, then all women should be married based on 1)
   - contradiction since this would mean all men are married

* Proof that this algorithm creates stable matches:
(Proof by contradiction)

- Let (m_1, w_1) and (m_2, w_2) be two married couples such that m_1 prefers w_2 to w_1 and w_2 prefers m_1 to m_2. This is an unstable match.
- Since m_1 prefers w_2 to w_1, w_2 must be higher than w_1 on his preference list. Similarly, m_1 must be higher than m_2 on w_2's preference list.
  - Then w_2 could not have divorced m_1 to marry m_2 ⇒ This is a contradiction.
    ⇒ There can be no unstable matches

∴ This algorithm always creates stable matches.

b) Example of a set of men + women w/ preference lists for which every perfect matching has a weak instability:

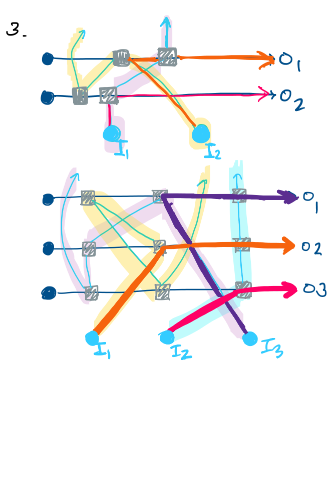


Let bar denote indifference with an adjacent barred element.

- w_1 prefers m_2, but m_2 is indifferent between her + his current wife
- w_2 prefers m_1, but m_1 is indifferent between her + his current wife
- w_3 prefers m_3, but m_3 is indifferent between her + his current wife

∴ There does not always exist a perfect matching with no weak instability

3.



Algorithm for input/output wires: runs in $O(n^2)$ time where n = number of input/output wires

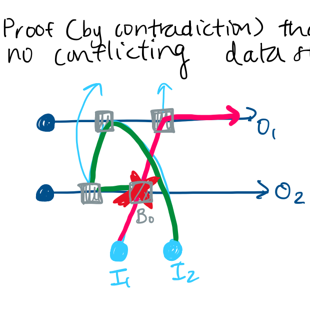
- While there is an output wire w/out a data stream
  - choose an arbitrary output wire O_0 w/out a data stream
  - while this output wire O_0 has no data stream
    - iterate through the junction boxes on O_0 starting from most downstream of the source to most upstream of the source
      - let the current junction box be denoted by J_0
      - if the input wire I_0 of J_0 is already supplying a data stream to a different output wire O_1 through junction box J_1
        ↳ if J_0 is upstream of J_1
          - turn off the switch for junction box J_1
          - turn on the switch for junction box J_0 so that I_0 sends its data stream to O_0

        ↳ else, move on to the next junction box
      - else
        ↳ switch J_0 on so that input wire I_0 sends its data stream onto O_0

① Proof (by contradiction) that this algorithm creates perfect matches between I/O wires:
- Let there be an output wire without a data stream and all input wires are sending data streams to their respective output wires at the end of the algorithm
- our algorithm only allows one data stream per output wire, so all output wires must have a data stream.
  ↳ This is a contradiction as we have an output wire w/out a data stream.

∴ There are always perfect matches such that every output wire has an input wire feeding its data stream to it.

② Proof (by contradiction) that this algorithm creates valid matches such that there are no conflicting data streams feeding into the same junction box:



- Suppose there is an invalid switching such that input I_1 is switched on to output O_1 and input I_2 is switched onto output O_2 such that both data streams pass through junction box B_0
- In order for this to happen, I_1 must be downstream of I_2 on O_2 and O_2 must be upstream of O_1 on I_1
- There are 2 possible cases:

  1) O_2 asked to receive I_1's data stream
     - Then I_1 switched to send its data stream to O_1 which means that O_1 is upstream of O_2

     ☆ This is a contradiction as we declared O_2 to be upstream of O_1 on I

  2) O_2 did not ask to receive I_1's data stream
     - Then O_2 asked to receive I_2's data stream before I_1, which means that I_2 is downstream of I_1.

     ☆ This is a contradiction as we declared I_1 to be downstream of I_2.

∴ There is always a valid switching of data streams.

4. 10^{10} operations per second, at most 1 hr of computation, find largest input size n

$$\frac{10^{10} \text{ operations}}{1 \text{ sec}} \times \frac{60 \text{ sec}}{1 \text{ min}} \times \frac{60}{\text{min}} = 3.6 \times 10^{13} \text{ operations}$$

a) n^2
$n^2 = 3.6 \times 10^{13}$
$\boxed{n = 6 \times 10^6}$

b) n^3
$n^3 = 3.6 \times 10^{13}$
$\boxed{n = 33019}$

c) $100n^2$
$100n^2 = 3.6 \times 10^{13}$
$\boxed{n = 6 \times 10^5}$

d) $n \log n$
$n \log n = 3.6 \times 10^{13}$
$\boxed{n \approx 9.06316 \times 10^{11}}$

e) 2^n
$2^n = 3.6 \times 10^{13}$
$\log_2 3.6 \times 10^{13} = n$
$\boxed{n = 45}$

f) 2^{2^n}
$2^{2^n} = 3.6 \times 10^{13}$
$\log_2 3.6 \times 10^{13} = 2^n$
$45 = 2^n$
$\log_2 45 = n$
$\boxed{n \approx 5}$

5. a) Prove by induction that the sum of the first n integers $(1+2+\ldots+n) = \frac{n(n+1)}{2}$

- Base case: Let $n=1$
  $1 = \frac{1(1+1)}{2}$
  $1 = 1$ ✓
- Inductive Hypothesis: Let $n=k$ such that $1+2+\ldots+k = \frac{k(k+1)}{2}$
- For $n=k+1$:

$$1+2+\ldots+k+(k+1) = \frac{k(k+1)}{2} + k+1 \quad \text{by inductive hypothesis}$$
$$= \frac{k(k+1) + 2(k+1)}{2}$$
$$= \frac{k^2+k+2k+2}{2}$$
$$= \frac{k^2+3k+2}{2}$$
$$= \frac{(k+1)(k+2)}{2}$$
$$= \frac{(k+1)((k+1)+1)}{2} \checkmark$$

∴ The sum of the first n integers $(1+2+\ldots+n) = \frac{n(n+1)}{2}$

b) Claim: $1^2+2^2+3^2+\ldots+n^2 = \frac{n(n+1)(2n+1)}{6}$

- Base case: $n=1$
  $1^2 = \frac{1(1+1)(2+1)}{6}$
  $1 = 1$ ✓
- Inductive hypothesis: Let $n=k$ such that $1^2+2^2+3^2+\ldots k^2 = \frac{k(k+1)(2k+1)}{6}$
- For $n=k+1$:

$$1^2+2^2+3^2+\ldots+k^2+\ldots(k+1)^2 = \frac{k(k+1)(2k+1)}{6} + (k+1)^2 \quad \text{by inductive hypothesis}$$
$$= \frac{k(k+1)(2k+1) + 6(k+1)^2}{6}$$
$$= \frac{(k+1)(2k^2+k+6(k+1))}{6}$$
$$= \frac{(k+1)(2k^2+7k+6)}{6}$$
$$= \frac{(k+1)(k+2)(2k+3)}{6}$$
$$= \frac{(k+1)((k+1)+1)(2(k+1)+1)}{6} \checkmark$$

∴ The sum of the first n consecutive squares $1^2+2^2+3^2+\ldots n^2 = \frac{n(n+1)(2n+1)}{6}$

6. Simpler example w/ 9 floors:

$O(n)$ one step at a time: 9
binary search: 6
$O(\sqrt{n})$ square root search: 5

	first egg breaks	total drops (worst case)	
floor	3	3	
	6	4	+1 drop
	9	5	+1 drop

Make total drops constant (not dependent which floor first egg breaks)
→ decrease intervals for first egg by 1 to keep drop # constant

Let x = number of drops (size of first interval)

For 9 floors:
$x + (x-1) + (x-2) + 1 = 9$
$+ \quad + 3 + 2 + 1 \approx 9$
→ $\frac{x(x+1)}{2} = 9$
$x^2 + x = 18$
$x^2 + x - 18 = 0$
$x = 3.772 \approx 4$ steps

For 200 floors:
$\frac{x(x+1)}{2} = 200$
$x(x+1) = 400$
$x^2 + x - 400 = 0$
$x = 19.506$
≈ 20 steps

When there are 200 steps, you need 20 drops in worst-case scenario.

Intervals for first egg: 0, 20, 39, 57, 74, 90, 105, 119, 132, 144, 155, 165, 174, 182, 189, 195, 200

20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5

- No matter where the first egg breaks, the total number of egg drops = 20.

* For a building with n floors, the worst case scenario for number of drops is x, where:

$$x + (x-1) + (x-2) + \ldots + 1 = n$$
$$\frac{x(x+1)}{2} = n$$
$$x^2 + x - 2n = 0$$
$$x = \frac{-1 + \sqrt{1-4(-2)n}}{2}$$

$\boxed{x = \frac{-1+\sqrt{1+8n}}{2} \text{ tries in the worst case when there are } n \text{ steps}}$

- Drop the first egg at floor x; interval count = 1
  - If it breaks, go to bottom-most floor in the current interval and test second egg until it breaks.
  - If it did not break, repeat process at (x minus interval count) floors above current floor.
    ↳ Increment interval count

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

This works because no matter what floor f_i the first egg breaks, the total drop count will be $i + (f_i - f_{i-1})$ where i is the interval count such that the first drop for the first egg is at interval $i=0$, keeping the max drop count constant no matter what floor the first egg breaks.

Runtime is $O(\sqrt{n})$.