WEEK 1 CS 180

ASHER CHRISTIAN 006-150-286

1. Day 1

1. famous
   a person is famous if every person in the room knows them and they no nobody.
   A know B? - allowed operation - yes or no
   optimize over the number of questions.
   worst case for n students to see if everyone knows one person need to ask n-1 questions.
   Then worst case we ask the person of interest n-1 questions to see if they know each person.
   $2(n-1)$ total questions.
   arbitrary vs random: random needs randomization (costly) (do not say randomize when mean arbitrary)
   repeat this process for checking famousness 2n(n-1) times but there can only be 1 or 0 famous people
   $2n^2 - 2n \sim n^2$
   There are n^2 relationships or n choose 2.
   If A knows B then A is not famous. If A does not know B then B is not famous.

$$
\begin{aligned}
Q&|n\\
1&|n-1\\
2&|n-2\\
&...\\
n-1&|1
\end{aligned}
$$

   And 2(n-1) to verify if the last person is famous. $3(n-1)$ questions.
   There must be at least n questions asked because otherwise there would be some person that we have no information from and which could change the outcome.

2. models of computation
   serial model of computation - von neumman model of computation
   A computer with an ALU and a local register - constant
   ALU can do basic operations (addition, subtraction, shift right, shift left) reasonable and simple
   Each operation takes about 1 unit of time
   We assume we have a large memory that we can read and write in one unit of time.

Date: 06.01.25.

Input output takes 1 unit of time.

3. $S = \sum_{i=1}^{n} x_i$
Input our numbers first into register then memory takes 2n units.
$x_1 \to ALU, x_2 \to ALU, +, x_3 \to ALU, +, ..., x_n \to ALU, +$
2(n-1) + 1
ouput + 1
4n total. $\sim n$.

2. Day 2

1. Matching problems
Given 2 groups, group A and group B with circles representing members of the group
We must match a member of A with a member of B according to some rule and restrictions specified by the problem
$|A| = n, |B| = n$. Every member of A must be matched with 1 and exactly 1 member of B

2. Stable Matching Problem
Match a_i with b_i? non-interesting.
Each element of A comes with a complete ranking or priority list
$$a_1 \to (3, 4, 1, 2, 6, ...).$$
The first element is the first choice, second is second choice, etc
For each b_i there is a complete ranking of A
If there exist a_i, a_j, b_k, b_l with $a_i - b_k$ and $a_j - b_l$ but a_i prefers b_l over b_k and b_k prefers a_i over a_j the matching is unstable. If for all such sets of pairs there are no unstable matches then the matching is stable.
By this definition there is always a stable configuration.

3. Algorithm to solve Stable Matching Problem as above
a_i asks b_k to match. If b_k has no match it says yes and (a_i, b_k) match.
If b_k has a match with a_j if $rank(j) > rank(i)$ then there is no match - reject the match
if $rank(i) > rank(j)$ b_k says yes and it drops a_j and (a_i, b_k) is now a match
Go to arbitrary a ask to match with the highest rank b not asked before. Ask to match with that b. b will accept if not matched. If it does have a match it will accept only if the rank of a is higher than its current match. There are n $a's$ and each a may ask at most n questions so there are at most n^2 operations and each operation takes one unit of time so the entire process runs in n^2 units of time.

4. Proof
Assume for contradiction that there exists an unstable matching. a matches with b and a' with b' but a prefers b' and b' prefers a. a must have asked b' in the past because b' is ranked higher than b for them.