DEF: LET US SAY THAT $b_j \in B$ IS A VALID MATCH FOR $a_i \in A$ IF $\exists$ STABLE MATCHING M S.T. $\{a_i, b_j\} \in M$.

DEF: LET $best(a_i)$, FOR $a_i \in A$, BE THE VALID MATCH $b^* \in B$ OF a_i, THAT a_i LIKES THE BEST.

THM A: THE G-S ALGORITHM RETURNS $M^* = \{\{a_i, best(a_i)\} \mid a_i \in A\}$

THM B: " " " " $M^* = \{\{b_j, worst(b_j)\} \mid b_j \in B\}$

P of THM A:

BY CONTRADICTION, SUPPOSE THAT SOME a_i ENDS UP BEING MATCHED TO A PARTNER OTHER THAN $best(a_i)$ IN M^*.

SINCE THE a'S PROPOSE IN DECREASING ORDER OF PREFERENCE, THERE MUST BE A TIME WHEN SOME a GETS REJECTED BY ONE OF HIS VALID MATCHES b (REJECTIONS CAN HAPPEN RIGHT AFTER A PROPOSAL, OR WHEN A b ACCEPTS SOME OTHER PROPOSAL).

LET THE PAIR $\{\bar{a}, \bar{b}\}$ BE THE FIRST PAIR (DURING THE EXECUTION OF G-S's ALGORITHM) THAT IS VALID, AND SUCH THAT $\bar{b}$ REJECTS $\bar{a}$.

WHEN THIS REJECTION HAPPENS, $\bar{b}$ WILL BE PAIRED UP WITH SOME $\bar{a}'$ THAT SHE LIKES BETTER THAN $\bar{a}$.

G-S's M^*

M'

NOW, SINCE $\{\bar{a}, \bar{b}\}$ IS A VALID PAIR THERE MUST EXIST A STABLE MATCHING M' SUCH THAT $\{\bar{a}, \bar{b}\} \in M'$.

SINCE M' IS A STABLE MATCHING, IT MUST MATCH $\bar{a}'$ TO SOME $\bar{b}'$, $\{\bar{a}', \bar{b}'\} \in M'$. THEN, $\{\bar{a}', \bar{b}'\}$ IS A VALID PAIR.

GIVEN THAT, $\{\bar{a}, \bar{b}\}$ WAS THE FIRST VALID PAIR WITH A REJECTION, IT MUST BE THAT $\bar{a}'$ WAS NOT REJECTED BY A VALID PARTNER BEFORE $\bar{a}'$ GETS ENGAGED WITH $\bar{b}$. (1)

$\bar{a}': \bar{b} > \bar{b}'$

$\bar{b}: \bar{a}' > \bar{a}$

NOW, SINCE $\bar{b}'$ IS A VALID PARTNER OF $\bar{a}'$ (2) $(\{\bar{a}', \bar{b}'\} \in M')$, AND SINCE $\bar{a}'$ PROPOSES IN DECREASING ORDER OF PREFERENCE (3), IT MUST BE THAT $\bar{a}'$ PREFERS $\bar{b}$ TO $\bar{b}'$.

SINCE $\bar{b}$ PREFERS $\bar{a}'$ TO $\bar{a}$ (SHE REJECTED $\bar{a}$ TO BE WITH $\bar{a}'$), AND SINCE $\{\bar{a}, \bar{b}\}, \{\bar{a}', \bar{b}'\} \in M'$, IT HOLDS THAT $\{\bar{a}, \bar{b}\}, \{\bar{a}', \bar{b}'\} \in M'$ IS AN INSTABILITY OF M'. THUS M' IS UNSTABLE. THIS IS A CONTRADICTION. $\square$

P of THM B:

SUPPOSE THAT $\exists \{\bar{a}, \bar{b}\} \in M^*$ S.T. $\bar{a} \neq worst(\bar{b})$.

$b: \bar{a} > \bar{a}'$

THEN, THERE EXISTS A STABLE MATCHING M' S.T. $\{\bar{b}, \bar{a}'\} \in M'$, AND $\bar{b}$ LIKES $\bar{a}'$ LESS THAN $\bar{a}$.

SUPPOSE THAT, IN M', $\bar{a}$ IS MATCHED TO SOME $\bar{b}' \neq \bar{b}$. $(\{\bar{a}, \bar{b}'\} \in M')$

BY THM. A, WE KNOW THAT $\bar{b}$ IS THE BEST VALID PARTNER OF $\bar{a}$.

THUS, $\bar{a}$ LIKES $\bar{b}$ MORE THAN $\bar{b}'$. (O/W THM. A WOULD NOT HOLD).

THAT IS, $\bar{a}: \bar{b} > \bar{b}'$ AND $\bar{b}: \bar{a} > \bar{a}'$.

BUT, $\{\bar{a}', \bar{b}\}, \{\bar{a}, \bar{b}'\} \in M'$. THUS, $\{\bar{a}', \bar{b}\}, \{\bar{a}, \bar{b}'\}$ FORM AN INSTABILITY OF M', WHICH IS THEN UNSTABLE. CONTRADICTION. $\square$

G-S HAS GIVEN US THE OPPORTUNITY TO DISCUSS A NUMBER OF NUMBER OF QUESTIONS ONE ENCOUNTERS WHEN STUDYING ALGORITHMIC PROBLEMS:

1. FORMULATED THE PROBLEM IN A MATHEMATICALLY PRECISE WAY;
2. ASK QUESTIONS ABOUT THE MATHEMATICAL PROBLEM.
3. DESIGN AN EFFICIENT ALG. FOR THE PROBLEM;
4. PROVE THAT YOUR ALGORITHM IS CORRECT AND BOUND ITS RUNTIME.

EFFICIENCY?

DEF (???): "AN ALGORITHM IS EFFICIENT IF, AFTER BEING IMPLEMENTED, IT RUNS QUICKLY ON REAL INPUT INSTANCES."

"QUICKLY": "QUICK" ON WHICH HARDWARE? WITH WHICH PROGRAMMING LANGUAGE? WITH WHICH IMPLEMENTATION?

WE WOULD LIKE OUR DEFINITION TO BE INDEPENDENT OF DETAILS.

"REAL INPUT INSTANCES": ... WHAT DOES IT MEAN? HOW TO DEFINE THEM?

TO PROVE SOMETHING SOMETHING ABOUT AN ALGORITHM WE NEED MATHEMATICAL DEFINITIONS.

WORST-CASE ANALYSIS

WE WANT THE RUNTIME OF OUR ALGORITHM TO BE BOUNDED IN TERMS OF THE WORST POSSIBLE INPUT.

FOR INSTANCE, IN THE CASE OF STABLE MATCHINGS, WE PROVED THAT G-S's ALGORITHM USES AT MOST n^2 ITERATIONS. $(n = |A| = |B|)$

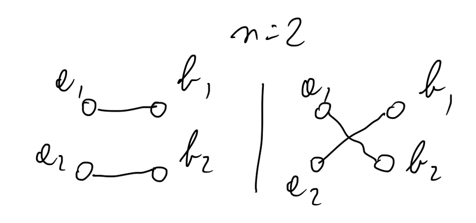
$a_1 : b_1 > \dots$
$a_2 : b_2 > \dots$
$\vdots$
$a_n : b_n > \dots$

SUPPOSE, FOR INSTANCE THAT WE CONSIDER INPUTS IN WHICH $|A| = |B| = N$, FOR $N = 2n$.

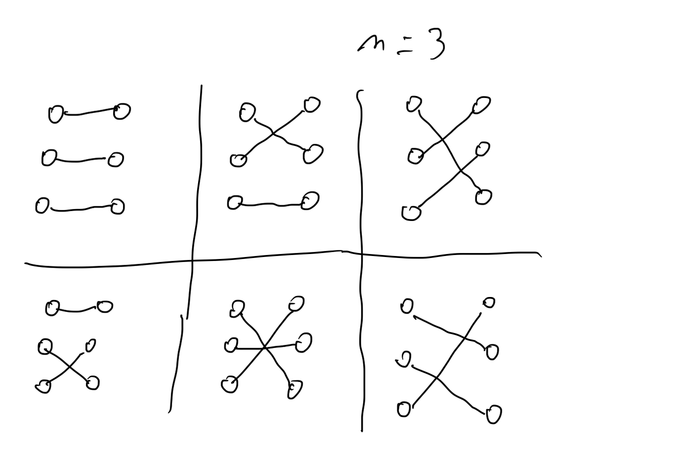
$$N^2 = (2n)^2 = 4n^2 \qquad n^2$$

DOUBLING n CHANGES THE RUNTIME BY A FACTOR 4.

THE BRUTE-FORCE ALGORITHM TRIED EACH OF THE $n!$ MATCHINGS:



$n! = 2! = 2$



$n! = 3! = 3 \cdot 2 \cdot 1 = 6$

WHAT WOULD HAPPEN TO THE BRUTE-FORCE ALGORITHM IF WE WERE TO DOUBLE THE NUMBER OF PEOPLE?

$$N! = (2n)! =$$
$$= 2n(2n-1)(2n-2)\cdots(n+1)\overbrace{n(n-1)\cdots 2 \cdot 1}$$
$$= 2n(2n-1)\cdots(n+1) \cdot n!$$
$$\geq (n+1)^n \cdot n! \geq 2^n \cdot n!$$

$n!$

GOOD DEF: AN ALGORITHM HAVING A RUNTIME $\leq c \cdot n^d$, WHERE c AND d ARE CONSTANTS, ON INPUTS OF SIZE n IS SAID TO BE A POLYNOMIAL TIME (POLYTIME) ALGORITHM.

IF I HAVE A POLYTIME ALGORITHM RUNNING IN TIME $c \cdot n^d$ ON INPUTS OF SIZE n, AND I RUN IT ON INPUTS OF SIZE $N = b \cdot n$,

$$cN^d = c(nb)^d = c\,n^d \cdot b^d \qquad c \cdot n^d$$

THEN THE BLOW-UP IN THE RUNTIME IS GOING TO BE A CONSTANT, b^d.