OFFICE HOURS TODAY
- 10 pm - 11 pm ET

Requests for extensions.

Stable Matching

Simpler algorithm

\[
\text{for each permutation } \pi \text{ of indices in } Y \ni
\]
\[
\text{Match } x_i \text{ with } \pi(i)
\]
\[
\text{check for any instability}
\]
\[
\text{if no instability then return the matching.}
\]

\(n! \quad n^2\)

Greatest Common Divisor

Def: let \(a, b \in \mathbb{Z}\). Then \(d\) is a
common divisor of \( a \) and \( b \), if
\[
\frac{d}{|a|} \text{ and } \frac{d}{|b|}.
\]

**Def.** Let \( a, b \in \mathbb{Z} \). Then \( d \) is
the greatest common divisor of \( a \) & \( b \).
i.e., \( d = \gcd(a, b) \)
- \( d \) is a common divisor of \( a \) & \( b \).
- \( d \) is the greatest common divisor of \( a \) & \( b \), \( d \geq e \).

**Algorithm.** \( \gcd(a, b) \)

\[
\text{for } k \leftarrow 1 \text{ to } \min(a, b) \text{ do }
\]
\[
\text{if } k \mid a \text{ and } k \mid b \text{ then }
\]
\[
\text{ans } \leftarrow k
\]

\[
\text{return } k
\]

**Lemma:** Let \( a \) and \( b \) be positive integers.
Let $c = a \mod b$. Then

$$\text{gcd}(a,b) = \text{gcd}(b,c).$$

\[
\begin{align*}
\text{gcd}(846, 315) &= \text{gcd}(315, 216) \\
&= \text{gcd}(216, 99) \\
&= \text{gcd}(99, 18) \\
&= \text{gcd}(18, 9) \\
&= 9
\end{align*}
\]

Proof of the lemma:

Let $d = \text{gcd}(a,b)$ and let $e = \text{gcd}(b,c)$.

We want to show that $d = e$.

$d \leq e$
We know that
\[ a = b \cdot q + c, \text{ for some } n \geq q. \]

\[ \therefore c = a - bq \]

Since \( d \mid a \) and \( d \mid b \), it must be that \( d \mid c \).

\[ \therefore d \mid b \text{ and } d \mid c. \]

But \( e = \gcd(b, c) \)

\[ \therefore d \leq e. \]

\[ e \leq d \]

\[ e \mid b, e \mid c \therefore e \mid a. \]

\[ \therefore e \mid a \text{ and } e \mid b. \]

But \( d = \gcd(a, b) \)
\[ d \geq e. \]

Euclid's alg \( \text{gcd}(a,b) \)

1. \( c \leftarrow a \mod b \)

2. \( \text{if } c = 0 \text{ then} \)
   
   \( \text{return } b \)

3. \( \text{else} \)
   
   \( \text{return } \text{gcd}(b,c) \).

\( \text{gcd}(315, 846) \)

\( = \text{gcd}(846, 315) \)

\[ \vdots \]

Correctness: Assume for contradiction that Euclid's alg fails. Among all pairs of
nos. on which Euclid's alg. fail, let $(a,b)$ be the pair s.t. $a+b$ is the smallest. Let $c = a \mod b$.

WLOG, let $a \geq b$.

Note that if $c = 0$ then our alg. returns $b$, which clearly is the correct ans. Here we assume $c > 0$.

\[
\begin{align*}
    c &< b \\
    + \quad b &\leq a \\
\hline
    b+c &< a+b
\end{align*}
\]

Since $c > 0$, Euclid's alg. calls $\gcd(b,c)$, which is correct (from
the lemma). Since \( \gcd(a, b) \) fails, it must be that \( \gcd(b, c) \) fails. But \( b + c < a + b \), contradicting that \( a + b \) is the smallest among all pairs \( (a, b) \) on which Euclid fails.

Lemma: Let \( a, b \in \mathbb{Z} \), such that \( a \geq b > 0 \). Let \( c = a \mod b \). Then \( c < a/2 \).

Proof: \( c < b \)

Case I: \( b < a/2 \)

Done.

Case II: \( b > a/2 \)
when \( a \) is divided by \( b \), the quotient is 1, & hence the remainder \( c = |a - b| < a/2 \).

\[
\begin{array}{r}
50 & \overset{1}{\overline{90}} \\
50 & \\
40 & (90 - 50)
\end{array}
\]

\[
\begin{array}{c}
a \\
\downarrow
\end{array}
\begin{array}{c}
b \\
\downarrow
\end{array}
\begin{array}{c}
c \\
\downarrow
\end{array}
\begin{array}{c}
d \\
\downarrow
\end{array}
\]

\[
\begin{array}{c}
< \frac{a}{2} \\
\downarrow
\end{array}
\begin{array}{c}
< \frac{b}{2} \\
\downarrow
\end{array}
\]

\[
(a, b) \xrightarrow{2 \text{ calls to gcd}} \left( < \frac{a}{2}, < \frac{b}{2} \right)
\]

\[
\xrightarrow{2 \text{ calls to gcd}} \left( < \frac{a}{4}, < \frac{b}{4} \right)
\]

\[
\xrightarrow{2 \text{ more calls to gcd}} \left( < \frac{a}{2^t}, < \frac{b}{2^t} \right)
\]

\[
\xrightarrow{\text{values in the last call to gcd}}
\]

\[
\# \text{ Calls} \leq 2.t
\]

\[
\frac{b}{2} \geq 1
\]
\[ 2^t \leq b \]
\[ \therefore t \leq \log_2 b \]

:. Total # times gcd is called \leq 2 \cdot \lg b.

Compare \( 2 \cdot \lg b \) to \( b \).

\[ b = 2^{1000} \]
\[ 2 \cdot \lg b = 2000 \]

**Sorting**

**Input:** array \( A[1..n] \) of distinct integers.

**Output:** the new sorted \( A \) in \( \Rightarrow \) order.

- Try all permutations
- check if the permutation is sorting order.
- if so, return it.

**InsulinSort** (A[1..n])

```
for j ← 2 to n do
    key ← A[j]
    i ← j-1

    while i > 0 and (A[i] > key) do
        A[i+1] ← A[i]
        i ← i-1
    A[i+1] ← key
```

```
A: 18 5 46 28
```

j = 2
key = 5
i = 1
InsertionSort (A[1..n])

for j ← 2 to n do
    key ← A[j]
    i ← j
    while A[i] > key do
        A[i] ← A[i - 1]
        i ← i - 1
    A[i] ← key

Cost: \[c_1 + \sum_{i=1}^{n} \text{#times} \leq n \times n\]

# times:
- \(c_1\): \(n\) times
- \(c_2\): \(n-1\) times
\[ \text{while } (i > 0) \land (A[i] > \text{key}) \text{ do} \]
\[ A[i+1] \leftarrow A[i] \]
\[ i \leftarrow i - 1 \]
\[ A[i+1] \leftarrow \text{key} \]

Total run time =
\[ C_1 \cdot n + C_2 (n-1) + C_3 (n-1) + C_4 \sum_{j=2}^{n} t_j + C_5 \sum_{j=2}^{n} (t_{j-1}) \]
\[ + C_6 \sum_{j=2}^{n} (t_{j-1}) + C_7 (n-1) \]

\[ = (C_4 + C_5 + C_7) n + (C_4 + C_5 + C_6) \sum_{j=2}^{n} t_j \]
\[ - (C_5 + C_6)(n-1) - (C_2 + C_3 + C_7) \]
= (C_1 + C_2 + C_3 + C_4 - C_5 - C_6) n + \sum_{j=1}^{n} t_j 
- (C_2 + C_3 + C_4 - C_5 - C_6)

**Best Case:** A is already sorted.

\[ t_j = 1 \]

\[ \therefore \text{Running time: } C_1 \cdot n + C_2 \cdot \sum_{j=2}^{n} 1 - C_3 \]

\[ = C_1 \cdot n + C_2 \cdot (n-1) - C_3 \]

\[ = an + b, \text{ when a, b are constants.} \]

\[ \sim n \]

**Worst Case:** Input is sorted in reverse order.

\[ t_j = j \]

\[ (\text{avg case } t_j = i/n^2) \]
Run time: \( C_1 \cdot n + C_2 \cdot \sum_{j=2}^{n} j - C_3 \)

\[ = C_1 \cdot n + C_2 \left( \frac{n(n+1)}{2} - 1 \right) - C_3 \]

\[ = C_1 \cdot n + C_2 \left( \frac{n^2}{2} + \frac{n}{2} - 1 \right) + C_3 \]

\[ \approx n^2 \text{ time.} \]

We are going to care about:

- worst case running time of the alg.
- running time of the alg as \( n \to \infty \) (asymptotic run time)
- When $n \to \infty$, we can ignore the lower order terms.