# Simple Greedy Algorithm

```
Approximate-Vertex-Cover(G=(V,E), k)
```

$C \leftarrow \emptyset$
$E' = G.E$
**while** $E' \neq \emptyset$ **do**
  $e = (u, v) \leftarrow$ arbitrary edge from $E'$
  $C = C \cup \{u, v\}$
  Remove from $E'$ every edge incident on $u$ or $v$
**return** $C$

- This does not achieve the minimum Vertex Cover.

## How bad can it be?

- Let $C$ be the cover returned
- Let $C^*$ be optimal cover
- Let $A$ be the set of edges chosen by algorithm

$$|A| \leq |C^*|$$

$$|C| = 2|A| \leq 2|C^*| \implies \frac{|C|}{|C^*|} \leq 2$$

2-approx

# Simple Greedy Algorithm

- Assign "next" job to least loaded machine
- Three machines, $\{\cancel{2}, \cancel{3}, \cancel{4}, \cancel{6}, \cancel{2}, \cancel{2}\}$?

$M_1$ 8 $T_1$
2 6
2 2 2

$M_2$ ~~3~~ 5 $T_2$
3 2
6

$M_3$ ~~4~~ 6 $T_3$
4 2
3 4

# Simple Greedy Algorithm

- Assign "next" job to least loaded machine
- Three machines, $\{\cancel{6}, \cancel{4}, \cancel{3}, \cancel{2}, \cancel{2}, \cancel{2}\}$?

| $M_1$ 6 | $M_2$ 6 | $M_3$ 7 |
|---|---|---|
| 6 | 4 | 3 |
| | 2 | 2 |
| | | 2 |

# Let's reason about OPT

$T = \max_i T_i$

$T^*$: OPT makespan

- Why is $T^*$ at least this?

$$T^* \geq \frac{1}{m} \sum_j t_j$$

equal division: $T^* = \frac{1}{m} \sum_j t_j$

else $T^* >$

# Let's reason about OPT

Why is it also true that $T^* \geq \max_j t_j$?

## How bad could our algorithm be?

$T_i$: $M_i$'s finish time

- What was last job placed on machine with max makespan?

Machine $i$, job $j$

at the time, $T_i - t_j$ was minimal

$$\sum_k t_k \geq m(T_i - t_j) \Rightarrow$$

$$T_i - t_j \leq \frac{1}{m}\sum t_k \leq T^*$$

$$+\ t_j \leq \max_k t_k \leq T^*$$

$$T_i \leq 2T^*$$

# How pessimistic is that?

2 - approx.. can we get 2 as worst case?

$m=3$: $M_1$ $M_2$ $M_3$ 1111113

ratio $5/3$

| $M_1$ | $M_2$ | $M_3$ |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 3 | | |

$m(m-1) \times 1$

$+ 1 \times m$

$$\Rightarrow \frac{m + m-1}{m} = \frac{2m-1}{m}$$

# What if we sort first?

$$\frac{1}{m}\sum_k t_k \le T^*$$

$m+1^{st}$ largest job?

$$2t_{m+1} \le T^* \Rightarrow t_{m+1} \le \frac{1}{2}T^*$$

$$T_i - t_j \le T^*$$

$$+ t_j \le t_{m+1} \le \frac{1}{2}T^*$$

$$T_i \le 1\tfrac{1}{2}T^*$$

1.5-approx

# Simple approximation

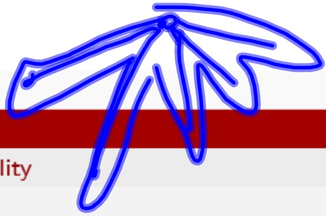
not true for all inputs

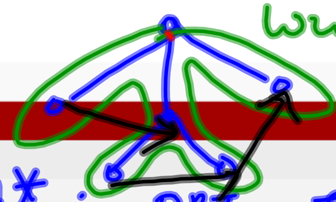
*Triangle Inequality*: $c(u,w) \leq c(u,v) + c(v,w)$

`Approximate-Traveling-Salesperson(G=(V,E))`

Select arbitrary $r \in V$ as starting point
$M \leftarrow \text{MST}(G)$
$T \leftarrow$ preorder traversal of $M$
**return** $T$

- Does this return a valid tour? Yes






$H^*$ : OPT TSP Tour

## Simple approximation

*Triangle Inequality*: $c(u,w) \leq c(u,v) + c(v,w)$

`Approximate-Traveling-Salesperson(G=(V,E))`

Select arbitrary $r \in V$ as starting point
$M \leftarrow \mathrm{MST}(G)$
$T \leftarrow$ preorder traversal of $M$
**return** $T$

- How bad could this be?

$c(M) \leq c(H^*)$

$W =$ walk on $M$

$c(W) = 2c(M)$
$\leq 2c(H^*)$

$c(T) \leq c(W)$ b/c tri ineq

$c(T) \leq 2 \cdot c(H^*)$

## How necessary is triangle inequality?

rho

- Suppose we have a $\rho$-approximate general TSP
- Use this to solve HAMILTONIAN CYCLE (G)

In G', OPT TSP?
n iff G has
$\geq n-1+\rho n$ otherwise

HC {
G' = complete graph on G.V

$$C_e = \begin{cases} 1 & e \in G \\ \rho n & e \notin G \end{cases}$$

c = $\rho$-approx TSP(G')
if c $\leq \rho n$
conclude G has HC
else
conclude G does not
}

# What does that mean?

- What does the previous prove about solving general TSP?

Just as NP-hard

Poly time $\rho$-approx T.S.P.

$\Rightarrow P = NP$

# Set Equality

1. For any set $S$, if $S$ can be put into 1:1 correspondence with the set $\{1, 2, \ldots n\}$, then we say $S$ has *cardinality* $n$. Two sets are the same size if they can be put into 1:1 correspondence.
2. The Schröder-Bernstein theorem states that if there exist injective functions $f : A \to B$ and $g : B \to A$, then there exists a bijective function $h : A \to B$

$f$: $|A| \leq |B|$

$g$: $|B| \leq |A|$

Let $\mathbb{N} = \{0, 1, 2, \ldots\}$. This is the set of *natural numbers*.
We say this set is *countably infinite* in size. We denote $|\mathbb{N}| = \aleph_0$.
Any set $S$ for which a bijective function $f : S \leftrightarrow \mathbb{N}$ exists is also countably infinite in size.

Let $\mathbb{N} = \{0, 1, 2, \ldots\}$. This is the set of *natural numbers*.
We say this set is *countably infinite* in size. We denote $|\mathbb{N}| = \aleph_0$.
Any set $S$ for which a bijective function $f : S \leftrightarrow \mathbb{N}$ exists is also countably infinite in size.

$\mathbb{N} \subset \mathbb{Z}$

- $\mathbb{Z} = \{\ldots -2, -1, 0, 1, 2 \ldots\}$, the set of all integers. How large is $\mathbb{Z}$?

$|\mathbb{N}| = |\mathbb{Z}|$

$f(2k) = k$

$f(2k+1) = -k$

$$f(n) = \frac{1 - (-1)^n(2n+1)}{4}$$

Let $\mathbb{N} = \{0, 1, 2, \ldots\}$. This is the set of *natural numbers*.
We say this set is *countably infinite* in size. We denote $|\mathbb{N}| = \aleph_0$.
Any set $S$ for which a bijective function $f : S \leftrightarrow \mathbb{N}$ exists is also countably infinite in size.

$E \subset N$

- Let $E$ be the set of *even* natural numbers (0, 2, 4, 6, etc). How large is $E$?

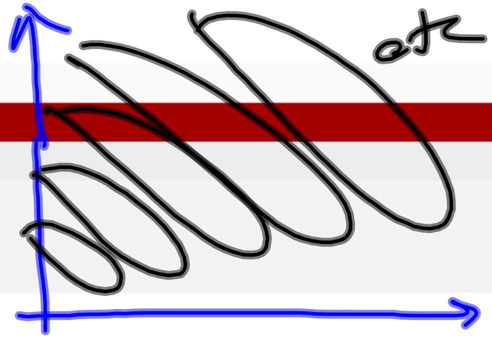
$f(x) = 2x$ $\quad f : N \to E$

# Rational Numbers

- $\mathbb{Q}$, the set of rational numbers; that is, all numbers that can be expressed as the ratio of two natural numbers. We can say that $\mathbb{Q} \subseteq \mathbb{N} \times \mathbb{N}$.
- Use Schröder-Bernstein to show that $|\mathbb{Q}| = |\mathbb{N}|$

$f: N \to Q \qquad f(x) = (x, 1)$

$g: Q \to N \qquad g(n, d) = 2^n \cdot 3^d$




# Rational Numbers

- $\mathbb{Q}$, the set of rational numbers; that is, all numbers that can be expressed as the ratio of two natural numbers. We can say that $\mathbb{Q} \subseteq \mathbb{N} \times \mathbb{N}$.
- Give an explicit bijection to show that $|\mathbb{Q}| = |\mathbb{N}|$

· partition $Q$ into $P_i = \{(n,d) \in Q \mid n+d == i\}$

· sort each $P_i$ by $x$ coord

# Rational Numbers

How many elements in $P_n$ appear before $(x, y)$?

if $(x,y)$ is in $P_n$,

$x$ elements appear before it

## Rational Numbers

How many total elements in $P_0, P_1, \ldots, P_{n-1}$?

How many $x, y \in \mathbb{N}$ sol'ns to:

$x + y < n$

$x + y + z = n - 1$

$n-1$ coins
3 buckets

$\left.\begin{array}{l} \\ \end{array}\right\}$ $n-1+3-1$ choose $3-1$

$= \binom{n+1}{2} = \frac{n(n+1)}{2}$

What # rational is $(x, y)$?

$\frac{n(n+1)}{2} + x$

$= \frac{(x+y)(x+y+1)}{2} + x$

# Rational Numbers

What's the bijection?

$$f(x,y) = \frac{(x+y)(x+y+1)}{2} + x$$

- Note that this is bijective between $\mathbb{Q}$ and $\mathbb{N}$.

$|N| = \aleph_0$

$|R| \overset{?}{=} \aleph_1 ?$

strictly >

$|R| > |N|$

# Real Numbers

- It's starting to look like all infinite sets are the same size
- What about $\mathbb{R}$, the set of real numbers?
- Suppose $|\mathbb{R}| = |\mathbb{N}|$.
  - Then there is a $f(x) : \mathbb{N} \to \mathbb{R}$

| $x$ | $f(x)$ |
|---|---|
| 0 | $\boxed{1}$. 3 |
| 1 | 3. $\boxed{1}$ 4 1 5 . . . . . |
| 2 | 6. 2 8 . . . . — |
| 3 | 2. 7 1 8 2 8 . . . . . |
| $\vdots$ | |

before dec: not same as f(0)

$i^{th}$ after dec $\neq$ f(i)'s $i^{th}$ after dec

both: if D = 6, 7
else 6

## The Barber Paradox

- There is a barber in a small village.
- The barber cuts the hair of exactly the villagers who do not cut their own hair.
- What is the paradox?
- Is this really a paradox?

# Computer Programs

- How many computer programs can exist? Does your answer change if we include programs that don't compile? **Hint**: Every computer program can be represented as a finite string with each character drawn from some character set.

  Countably infinite

- How many possible inputs are there to computer programs?

  countably infinite

# Let's define behavior...

| | $x_0$ | $x_1$ | $x_2$ | $x_3$ | $\dots$ |
|---|---|---|---|---|---|
| $P_0$ | F | T | F | F | $\dots$ |
| $P_1$ | T | T | T | F | $\dots$ |
| $P_2$ | T | T | F | F | $\dots$ |
| $P_3$ | F | F | F | F | $\dots$ |
| $P_4$ | T | F | F | T | $\dots$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ |

- Table entry $(i, j)$ is true if:
  - $P_i$ is a program that compiles and runs,
  - AND $P_i$ enters into an infinite loop if $x_j$ is provided as input.
- It's false otherwise
  - $P_i$ doesn't compile
  - OR $P_i$ has finite running time

does it halt?

## Halting

- Let's define `Halts`$(P, x)$ to be a boolean function that returns true or false according to whether or not program $P$ will eventually halt, given $x$ as input.
- A return value of "true" indicates that yes, the program will halt.
- Suppose this can be implemented. Are there any programs that cannot be?

Q (Xi) ← impossible

find Pi, xi ← not the impossible part

the impossible part

if (Halts (Pi, Xi))

return ~~false~~ true

else

return ~~true~~ false