Traveling Salesperson Problem (TSP)

Algorithm running time

By the end of this video you will be able to…

- Analyze the running time of the brute-force and the greedy algorithms for the TSP

In TSP, given n cities with one Hometown and all pairwise distances, plan a tour starting and ending at Hometown that visits every city exactly once and has minimum distance.

Brute force algorithm: Generate all paths and choose the shortest

SD → Lima → Paris → Cairo → Perth → Beijing → Johannesburg → Chennai → San Diego

6,091 + 10,248 + 3210 + 11,258 + 7,985 + 11,699 + 7,133 + 14,587 = 72,211km

SD → Lima → Paris → Cairo → Perth → Beijing → Chennai → Johannesburg → San Diego

6,091 + 10,248 + 3210 + 11,258 + 7,985 + 4,615 + 7,133 + 16,575 = 67,115km

SD → Lima → Paris → Cairo → Perth → Johannesburg → Beijing → Chennai → San Diego

6,091 + 10,248 + 3210 + 11,258 + 8,308 + 11,699 + 4,615 + 14,587 = 70,016km

. . .

In TSP, given n cities with one Hometown and all pairwise distances, plan a tour starting and ending at Hometown that visits every city exactly once and has minimum distance.

Brute force algorithm: Generate all paths and choose the shortest

SD → Lima → Paris → Cairo → Perth → Beijing → Jo urg → Chennai → San Diego

6,091 + 10,248 + 3210 + 11,258 + 7,9 7,133 + 14,587 = 72,211km

SD → Lima → Paris → Cai hennai → Johannesburg → San Diego

6,091 + 10, ,985 + 4,615 + 7,133 + 16,575 = 67,115km

SD → Lima → airo → Perth → Johannesburg → Beijing → Chennai → San Diego

6,091 + 10,248 + 3210 + 11,258 + 8,308 + 11,699 + 4,615 + 14,587 = 70,016km

. . .

It works!!
(But how long does it take…)

In TSP, given n cities with one Hometown and all pairwise distances, plan a tour starting and ending at Hometown that visits every city exactly once and has minimum distance.

Brute force algorithm: Generate all paths and choose the shortest

But how many permutations?!?

```
bestPath = null, bestDist = +Infinity
for each permutation of cities starting and ending in Hometown:
    calculate distance of current permutation
    if (distance < bestDist)
        bestPath = current permutation, bestDist = distance

return bestPath
```

$O(n)$

In TSP, given n cities with one Hometown and all pairwise distances, plan a tour starting and ending at Hometown that visits every city exactly once and has minimum distance.

Brute force algorithm: Generate all paths and choose the shortest

How many permutations?

San Diego
Cairo
Johannesburg
Chennai
Lima
Paris
Beijing
Perth

In TSP, given n cities with one Hometown and all pairwise distances, plan a tour starting and ending at Hometown that visits every city exactly once and has minimum distance.

Brute force algorithm: Generate all paths and choose the shortest

How many permutations?

San Diego
Cairo
Johannesburg
Chennai
Lima
Paris
Beijing
Perth

<IVQ placeholder>

How many permutations are there for this tour?

A. 7!
B. 7^n
C. 2^7
D. 2*8

In TSP, given n cities with one Hometown and all pairwise distances, plan a tour starting and ending at Hometown that visits every city exactly once and has minimum distance.

Brute force algorithm: Generate all paths and choose the shortest

How many permutations?

San Diego
Cairo
Johannesburg
Chennai
Lima
Paris
Beijing
Perth

How many choices for the first city?

In TSP, given n cities with one Hometown and all pairwise distances, plan a tour starting and ending at Hometown that visits every city exactly once and has minimum distance.

Brute force algorithm: Generate all paths and choose the shortest

How many permutations?

San Diego
Cairo
Johannesburg
Chennai
Lima
Paris
Beijing
Perth

How many choices for the first city? 1 (San Diego)

In TSP, given n cities with one Hometown and all pairwise distances, plan a tour starting and ending at Hometown that visits every city exactly once and has minimum distance.

Brute force algorithm: Generate all paths and choose the shortest

How many permutations?

~~San Diego~~
Cairo
Johannesburg
Chennai
Lima
Paris
Beijing
Perth

How many choices for the first city? 1 (San Diego)
How many choices for the next city?

In TSP, given n cities with one Hometown and all pairwise distances, plan a tour starting and ending at Hometown that visits every city exactly once and has minimum distance.

Brute force algorithm: Generate all paths and choose the shortest

How many permutations?

~~San Diego~~
~~Cairo~~
Johannesburg
Chennai
Lima
Paris
Beijing
Perth

How many choices for the first city? 1 (San Diego)
How many choices for the next city? 7

In TSP, given n cities with one Hometown and all pairwise distances, plan a tour starting and ending at Hometown that visits every city exactly once and has minimum distance.

Brute force algorithm: Generate all paths and choose the shortest

How many permutations?

~~San Diego~~
~~Cairo~~
Johannesburg
Chennai
Lima
Paris
Beijing
Perth

How many choices for the first city? 1 (San Diego)
How many choices for the next city? 7
How many choices for the next city?

In TSP, given n cities with one Hometown and all pairwise distances, plan a tour starting and ending at Hometown that visits every city exactly once and has minimum distance.

Brute force algorithm: Generate all paths and choose the shortest

How many permutations?

~~San Diego~~
~~Cairo~~
Johannesburg
Chennai
Lima
Paris
Beijing
Perth

How many choices for the first city? 1 (San Diego)
How many choices for the next city? 7
How many choices for the next city? 6

For each of the first 7 cities, we can choose 6 different next cities.
So far we have 7*6 = 42 different paths started

In TSP, given n cities with one Hometown and all pairwise distances, plan a tour starting and ending at Hometown that visits every city exactly once and has minimum distance.

Brute force algorithm: Generate all paths and choose the shortest

How many permutations?

~~San Diego~~
~~Cairo~~
Johannesburg
Chennai
Lima
Paris
Beijing
Perth

How many choices for the first city? 1 (San Diego)
How many choices for the next city? 7
How many choices for the next city? 6
How many choices for the next city? 5
How many choices for the next city? 4
How many choices for the next city? 3
How many choices for the next city? 2
How many choices for the next city? 1
How many choices for the last city? 1 (San Diego)

So overall we have 1*7*6*5*4*3*2*1*1 = 5040 paths = 7!

In TSP, given n cities with one Hometown and all pairwise distances, plan a tour starting and ending at Hometown that visits every city exactly once and has minimum distance.

Brute force algorithm: Generate all paths and choose the shortest

How many permutations?

~~San Diego~~
~~Cairo~~
Johannesburg
Chennai
Lima
Paris
Beijing
Perth

How many choices for the first city? 1 (San Diego)
How many choices for the next city? 7
How many choices for the next city? 6
How many choices for the next city? 5
How many choices for the next city? 4
How many choices for the next city? 3
How many choices for the next city? 2
How many choices for the next city? 1
How many choices for the last city? 1 (San Diego)

In general we have (n-1)! permutations to try!

In TSP, given n cities with one Hometown and all pairwise distances, plan a tour starting and ending at Hometown that visits every city exactly once and has minimum distance.

Brute force algorithm: Generate all paths and choose the shortest

(n-1)! permutations

```
bestPath = null, bestDist = +Infinity
for each permutation of cities starting and ending in Hometown:
    calculate distance of current permutation
    if (distance < bestDist)
        bestPath = current permutation, bestDist = distance

return bestPath
```

O(n)

$(n-1)! * n = O(n!)$

N	N!
10	~3.6 million
19	1.22 x 10^{17} (the age of the universe)
23	# of stars in the universe
59	# of atoms in the universe

In TSP, given n cities with one Hometown and all pairwise distances, plan a tour starting and ending at Hometown that visits every city exactly once and has minimum distance.

Greedy algorithm: pick best next choice

In TSP, given n cities with one Hometown and all pairwise distances, plan a tour starting and ending at Hometown that visits every city exactly once and has minimum distance.

Greedy algorithm: pick best next choice

```
bestPath = []
current = Hometown
cities to visit = all other cities
while (more cities to visit)
    select city closest to current and add to bestPath
    remove current city from cities to visit
    current = selected city
return bestPath
```

n-1 cities

How long does this take?

In TSP, given n cities with one Hometown and all pairwise distances, plan a tour starting and ending at Hometown that visits every city exactly once and has minimum distance.

Greedy algorithm: pick best next choice

```
bestPath = []
current = Hometown
cities to visit = all other cities
while (more cities to visit)
    select city closest to current and add to bestPath
    remove current city from cities to visit
    current = selected city
return bestPath
```

n-1 cities

How long does this take? O(n)

In TSP, given n cities with one Hometown and all pairwise distances, plan a tour starting and ending at Hometown that visits every city exactly once and has minimum distance.

Greedy algorithm: pick best next choice

```
bestPath = []
current = Hometown
cities to visit = all other cities
while (more cities to visit)
    select city closest to current and add to bestPath
    remove current city from cities to visit
    current = selected city
return bestPath
```

n-1 cities

How long does this take? O(n)

Total algorithm takes n-1*n = $O(n^2)$