

Cheapest link algorithm.

Last time we saw an algorithm that gives a non optimal solution of the traveling salesman pb but works in practice. The idea was to consider the nearest neighbor, but we had no choice once we chose where to start.

Is there a better algorithm? There is another one, but it won't be better nor worse.

Traveling salesman pb: NP hard problem.

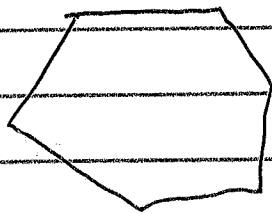
solvable in polynomial time by a non deterministic Turing machine.

Belief = we will not be able to solve the TSP  
without going through all the solutions.

Difficulty is finding a path!!

Cheapest link algorithm

- Find the cheapest edge
- avoid 2 situations

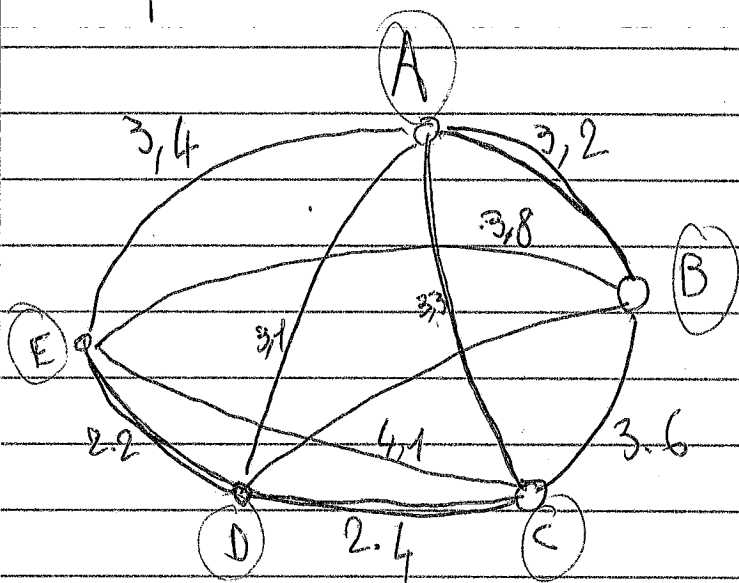


and

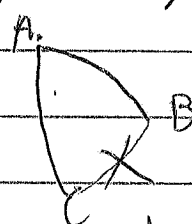
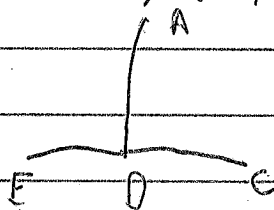


Continue until you get a path.

Example: A



cheapest = ED, DC, ~~DA~~, AB, AC, ~~BC~~



circle the vertices of the edges we chose. Forbidden.

Cost:  $2.2 + 2.4 + 3.3 + 3.2 = 11.1$

# Example B

	A	B	C	D	E
A	*	1.2	0.7	1.0	1.3
B		*			
C			*		
D				*	
E					*

	(A)	(B)	(C)	(D)	(E)
A	*	3.2	3.3	3.1	3.4
B	3.2	*	3.6	<del>2.4</del> 4.5	3.8
C	3.3	3.6	*	2.4	4.1
D	3.1	4.5	2.4	*	2.2
E	3.4	3.8	4.1	2.2	*

Cheapest link: ED, DC, ~~DA~~, BA, AC stop.

$$\text{check } \sum \text{edges} - 2 \times \text{vertices}$$

$$\sum \text{deg vert} = 2 \times \text{edges}$$