AUSTRALIAN MATHS TRUST

Algorithms

Workbook

critical thinking, creative solutions

SOLVE PROBLEMS. CREATE THE FUTURE.

Algorithms

Part 1 – A whirlwind tour of Big Deal™ Algorithms.

1. Highest common factor: **Euclid's algorithm**, Alexandria (Egypt), c. 300 BC
2. Prime number?: **The sieve of Eratosthenes**, Alexandria (Egypt), c. 300 BC
3. Shortest path: **Dijkstra's algorithm**, Amsterdam (Holland), 1959
4. Sorting: **Mergesort**, Princeton (US), 1945
5. Searching a list: **linear search, binary search**
6. Difficult planning: **Travelling Salesman**, UCB (US), 1972
7. Analysing the Web: **PageRank algorithm**, Stanford/Google (US), 1998
8. Representing 3D graphics: **Oct trees**, New York (US), 1980
9. Deep learning: **Modern neural networks**

Part 2 – Algorithms and scale.

1. Time complexity
2. Is P = NP?

Highest common factor: **Euclid's algorithm**, part 1

This is your chance to have a go at finding the answer BEFORE you hear about how the algorithm works. If you already know the algorithm, you are of course most welcome to use it!

Determine hcf(4,48), the highest common factor of 4 and 48.

Determine hcf(138,60), the highest common factor of 138 and 60.

Determine hcf(360,3402), the highest common factor of 360 and 3402.

Highest Common Factor: **Euclid's algorithm** part 2

Take notes here:

Use Euclid's algorithm to determine hcf(32,88), the highest common factor of 32 and 88.

Use Euclid's algorithm to determine (again!) hcf(360,3402), the highest common factor of 360 and 3402.

Prime number?: **The sieve of Eratosthenes** part 1

This is your chance to have a go at finding the answer BEFORE you hear about how the algorithm works. If you already know the algorithm, you are of course most welcome to use it!

1. Is 43 a prime number?
2. Is 91 a prime number?
3. Is 109 a prime number?

The following table may be useful:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110

Prime number?: **The sieve of Eratosthenes** part 2

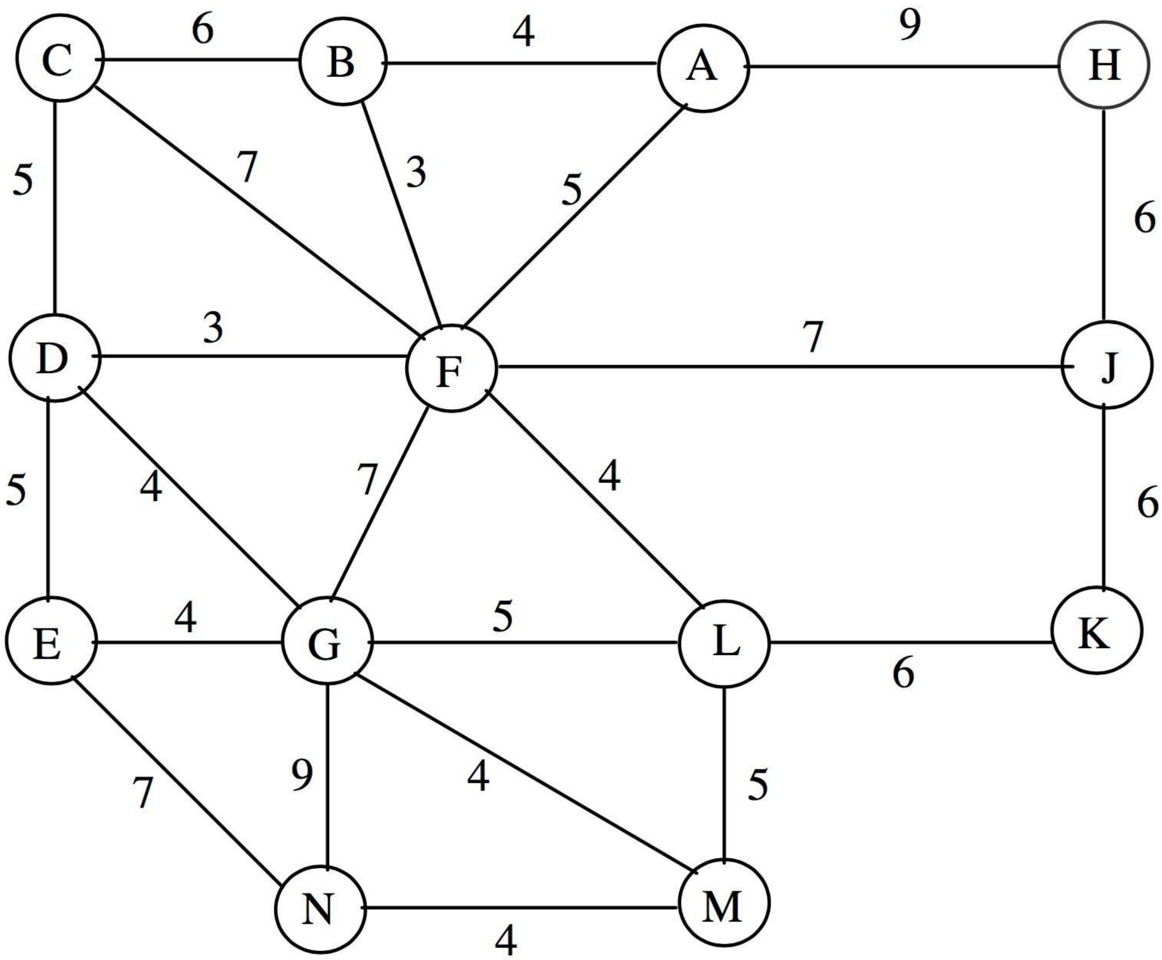
Now listen carefully and take notes!

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110

Shortest path: **Dijkstra's algorithm** part 1

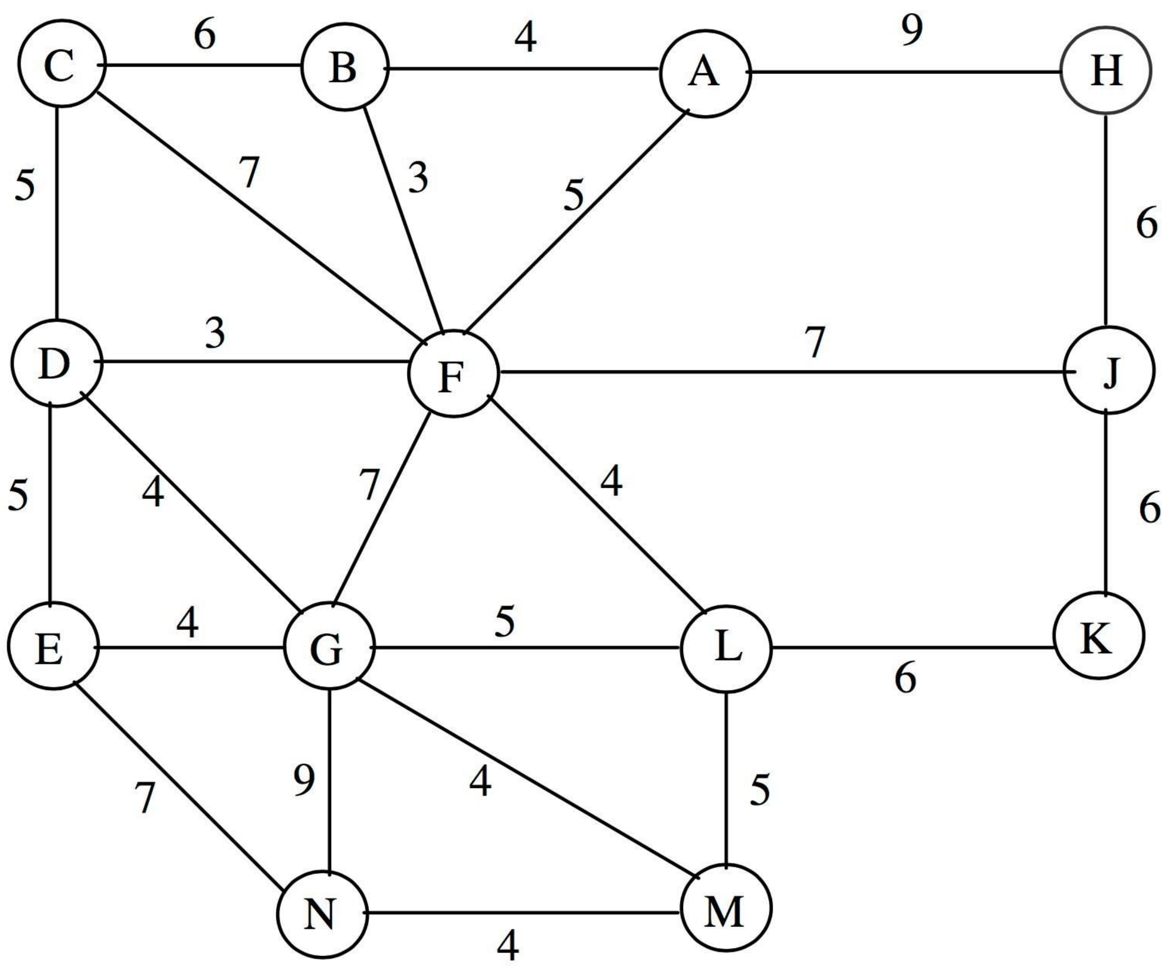
This is your chance to have a go at finding the answer BEFORE you hear about how the algorithm works. If you already know the algorithm, you are of course most welcome to use it!

Find the shortest path from E to H in the network below. The numbers on each connecting line denote travel time (in km).



Shortest path: **Dijkstra's algorithm** part 2

Now listen carefully and take notes!



Sorting: **Mergesort** part 1

This is your chance to have a go at finding the answer BEFORE you hear about how the algorithm works. If you already know the algorithm, you are of course most welcome to use it!

(a) How would you as a computer sort the numbers 6 5 3 1 9 8 2 4 ?

(b) What if you were part of a *team* of computers, working in parallel and combining your results?

Sorting: **Mergesort** part 2

Now listen carefully and take notes!

Searching a list: **linear search**, **binary search** part 1

This is your chance to have a go at finding the answer BEFORE you hear about how the algorithm works. If you already know the algorithm, you are of course most welcome to use it!

a. How would you, as a computer, check if the number 7 is in the following list?

2 4 6 3 23 1 43 2 6 98 9 8 63 2 14 12 7 41 9 5

b. How would you, as a computer, check if the number 7 is in the following list?

1 2 2 2 3 4 5 6 6 7 8 9 9 12 14 23 41 43 63 98

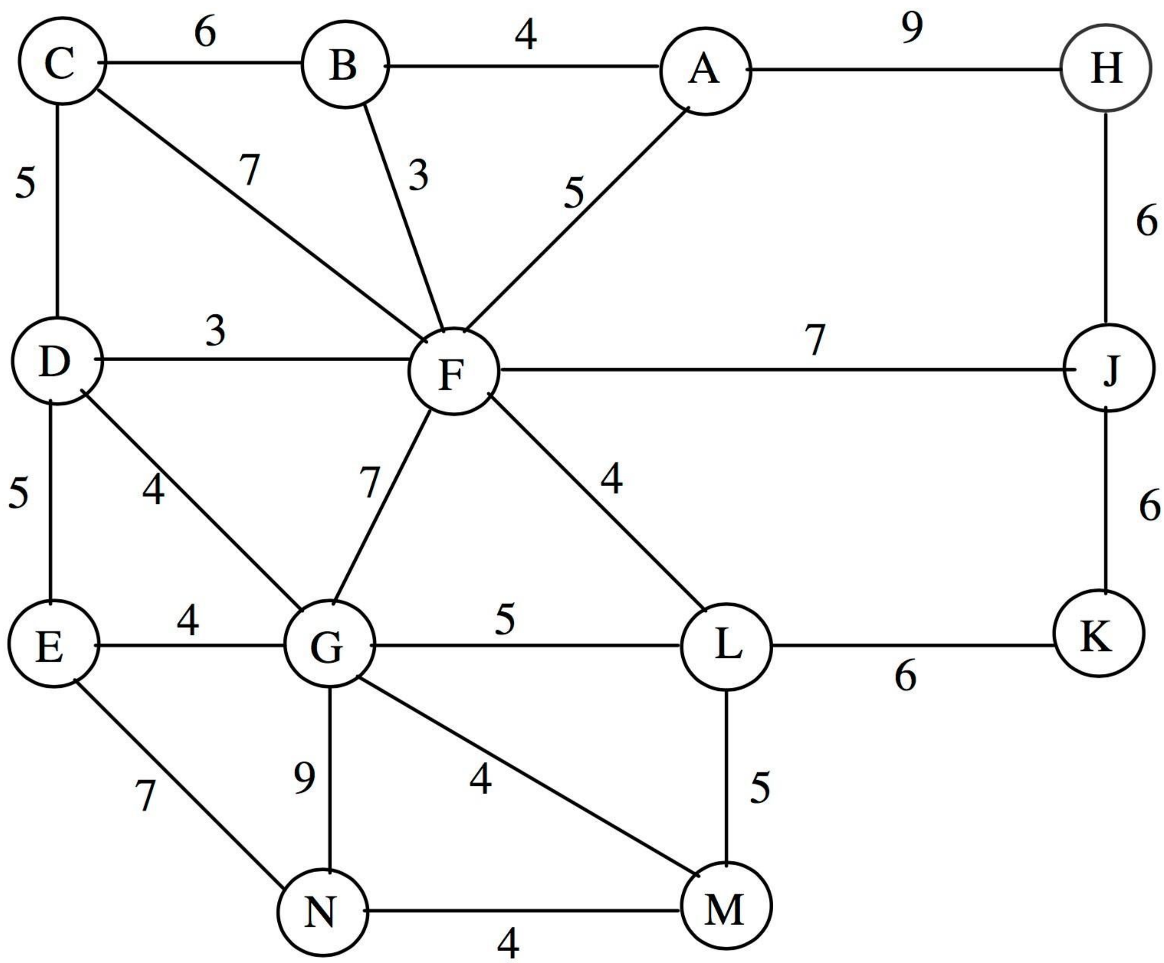
Searching a list: **linear search**, **binary search** part 2

Now listen carefully and take notes!

The travelling salesman problem
part 1

This is your chance to have a go at finding the answer BEFORE you hear about how the algorithm works. If you already know the algorithm, you are of course most welcome to use it!

Find the shortest path that visits each node exactly once for the network below. (The numbers denote distances in km.)



The travelling salesman problem
part 2

Now listen carefully and take notes!

Analysing the Web: Google's **PageRank algorithm**

Now listen carefully and take notes!

Representing 3D graphics: **Oct trees**

Now listen carefully and take notes!

Deep learning: **Modern Neural networks**

Now listen carefully and take notes!

Complexity of algorithms

Time complexity discussion

Time complexity of Euclid's algorithm: ___________

Time complexity of the sieve of Eratosthenes: ___________

Time complexity of Dijkstra's algorithm: ___________

Time complexity of Kruskal's algorithm: ___________

Time complexity of Mergesort: ___________

Time complexity of linear search: ___________

Time complexity of binary search: ___________

Time complexity of travelling salesman problem: ___________

Comparing complexities

n	1	2	3	4	5	6
log(n)	0	0.30	0.48	0.60	0.70	0.78
n	1	2	3	4	5	6
n*log(n)	0	0.60	1.43	2.41	3.49	4.67
n^2	1	4	9	16	25	36
n^3	1	8	27	64	125	216
n!	1	2	6	24	120	720
n^n^n	1	16	19683	4294967296	298023223876953000	10314424798490500000000000000

Finally, a $1,000,000 question: Does "P = NP"?

Now listen carefully and take notes!

See also:

- http://www.claymath.org/millennium-problems
- "What Does 'P vs. NP' Mean for the Rest of Us?" http://bit.ly/2Dbhumh (2010)