BHASVIC
MαTHS

D1 Survival Kit

Decision 1

Matchings
Bin Packing, Binary Search
Bubble Sort, Quick Sort
Dijkstra
Kruskal, Prim
Linear Programming
Route Inspection
Critical Path Analysis
Gantt Charts
Scheduling
Glossary

Name ....

Matchings:

Start:

Alternating Path/Change status – *means* not in current matching

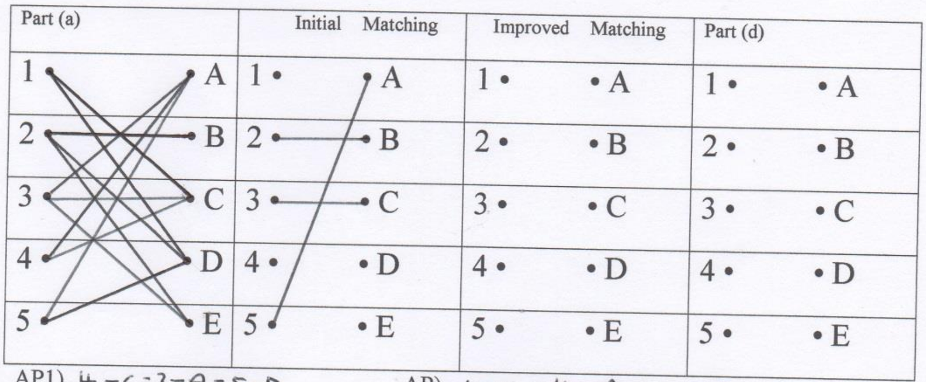
= *means* in current matching

End:

A college has 5 vacant jobs, A, B, C, D, E, and 5 applicants, 1, 2, 3, 4, 5.
The applicants are only qualified for certain jobs as is shown in the table:

Applicant	Jobs qualified for
1	C, D
2	B, D, E
3	A, C, E
4	A, C
5	A, D

a) Show this information on a bipartite graph.
b) Initially applicant 2 is matched to job B, 3 to C and 5 to A. Using this initial matching, find 3 distinct alternating paths.
c) Using the longest of your alternating paths, use the maximal matching algorithm to obtain an improved matching and hence obtain a complete matching.
d) The interview panel decide that applicant 3 will be appointed to job C. Explain why this means that it is not possible to fill the remaining jobs with the remaining applicants.



AP1) 4 – C = 3 – A = 5 – D

AP2) 4 – C = 3 – D

AP3) 4 – A = 5 – D

CS) 4 = C – 3 = A – 5 = D

Improved Matching:

1 = unmatched

2 = B

3 = A

4 = C

5 = D

AP) 1 – C = 4 – A = 3 – E

CS) 1 = C – 4 = A – 3 = E

Complete Matching:

1 = C

2 = B

3 = E

4 = A

5 = D

Part d) applicants 1, 4 and 5 can only do jobs A, C and D. So if 3 does C, one of 1, 4 and 5 has no suitable job

Bin Packing, Binary Search

Each of 3 storage crates has a 12 cubic metre capacity. There are various boxes of the following sizes A(2) B(8) C(3) D(7) E(5) F(3) G(3) H(4) I(2)

What is the minimum number of crates needed?

total = 37 ∴ minimum = $\frac{37}{12} = 3\frac{1}{12}$, i.e. 4

(a) Show how these boxes can be packed in a minimum number of crates by first fit

2	3	5	4
8	7	3	
2		3	

(b) Show how these boxes can be packed in a minimum number of crates by first fit decreasing B(8) D(7) E(5) H(4) C(3) F(3) G(3) A(2) I(2)

8	7	5	2
4	3	3	
	2	3	

CHECK CHECK CHECK CHECK CHECK CHECK CHECK

Use the binary search algorithm to locate 'Paul' in this list:

1 Alison
2 Bobby
3 Brian
4 Graham
5 Jeremy
6 Malcolm
7 Paula
8 Sarah
9 Tom

middle no. = 5th = Jeremy
Paul is after Jeremy
Discard 1–5

middle no. = 8th = Sarah
Paul is before Sarah
Discard 8–9

middle no. = 7th = Paula
Paul is before Paula
Discard 7

one remaining name = Malcolm
Malcolm is not Paul
∴ Paul not found
Search complete

3

Bubble/Quick Sort

Bubble

2	1	11	3	4	3	8	5
1	2	11	3	4	3	8	5
1	2	11	3	4	3	8	5
1	2	3	11	4	3	8	5
1	2	3	4	11	3	8	5
1	2	3	4	3	11	8	5
1	2	3	4	3	8	11	5
1	2	3	4	3	8	5	11
1	2	3	3	4	5	8	11
1	2	3	3	4	5	8	11

sort complete

Quick

2 1 11 3 $\boxed{4}$ 3 8 5

2 1 $\boxed{3}$ 3 4 11 $\boxed{8}$ 5

2 $\boxed{1}$ 3 3 4 5 8 11

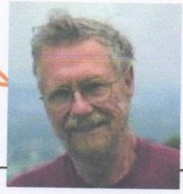
1 2 3 3 4 5 8 11

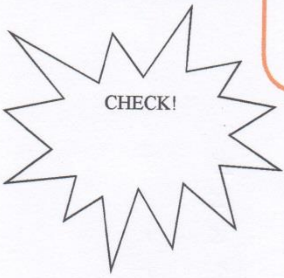
sort complete

4

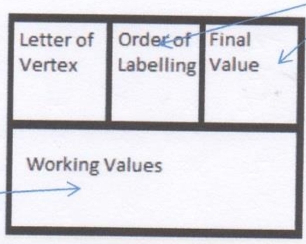
Dijkstra

Hi, I'm Professor Dijksta. You can use my algorithm to find:



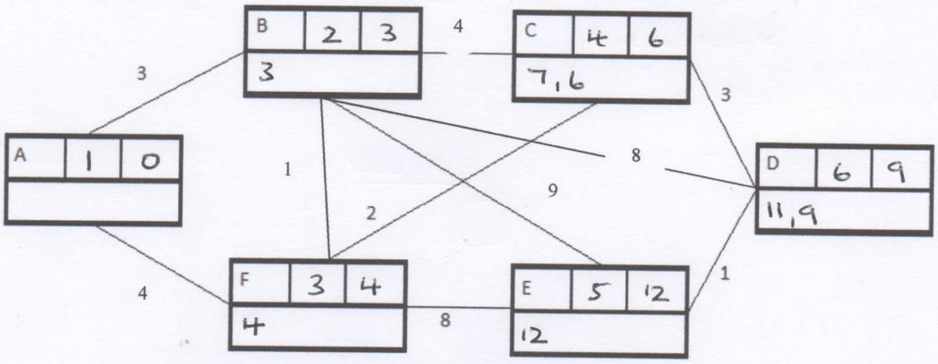


(1)

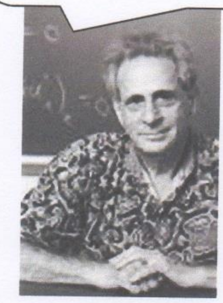


(2)

Find the shortest distance from A to E and a route that gives this distance



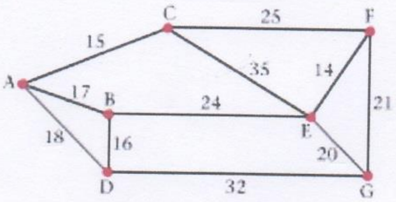
Kruskal & Prims:

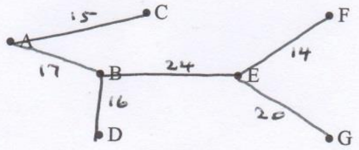


Key Differences

Kruskal

EF = 14 accept
AC = 15 accept
BD = 16 accept
AB = 17 accept
AD = 18 reject
EG = 20 accept
FG = 21 reject
BE = 24 accept
CF = 25 reject
DG = 32 reject
CE = 35 reject



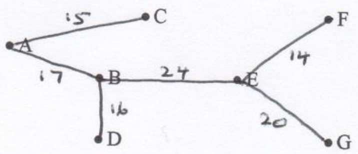


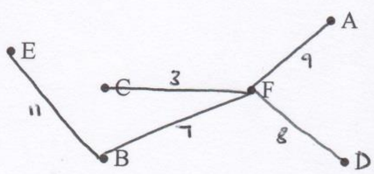
Weight = 14 + 15 + 16 + 17 + 20 + 24 = 106 units

	1 A	4 B	3 C	5 D	6 E	2 F
A	~~-~~	~~10~~	~~12~~	~~13~~	~~20~~	~~9~~
B	~~10~~	~~-~~	~~7~~	~~15~~	~~11~~	7
C	~~12~~	~~7~~	~~-~~	~~11~~	~~18~~	3
D	~~13~~	~~15~~	~~11~~	~~-~~	~~27~~	8
E	~~20~~	11	~~18~~	~~27~~	~~-~~	~~18~~
F	9	~~7~~	~~3~~	~~8~~	~~18~~	~~-~~

Prim at A

AC
AB
BD
BE
EF
EG





Linear Programming:

The young enterprise company 'Decide' is going to produce badges to sell to decision maths students. It will produce two types of badges. Badge 1 reads 'I made the decision to do maths' and badge 2 reads 'Maths is the right decision'. They will sell the badges for 30p (badge 1) and 40p (badge 2) and wish to maximise income.

DEFINE: x = badge 1, y = badge 2 **OBJECTIVE:** Maximise income

'Decide' must produce at least 200 badges and have enough material for 500 badges.

$x+y \geq 200$ $x+y \leq 500$

Market research suggests that the number produced of badge 1 should be between 20% and 40% of the total number of badges made.

$x > 0{\cdot}2\,(x+y)$ $\Rightarrow 0{\cdot}8x > 0{\cdot}2y \Rightarrow 4x > y$

$x < 0{\cdot}4\,(x+y)$, $0{\cdot}6x < 0{\cdot}4y$ $\therefore 3x < 2y$

Type	No.	cost
1	x	30
2	y	40

'Decide' want to sell at most twice as many badge 1 as badge 2.

$2y \geq x$

Don't forget: $x \geq 0$, $y \geq 0$

Show the following constraints on a graph and label the feasible region

$2x+3y<12$

$y \geq 2-x$

$x \geq 0$

$y \geq 0$

$2y > x$

$2y-5x \leq 0$

Label each line!

$2x+3y=12$

x	y
0	4
6	0

$y=2-x$

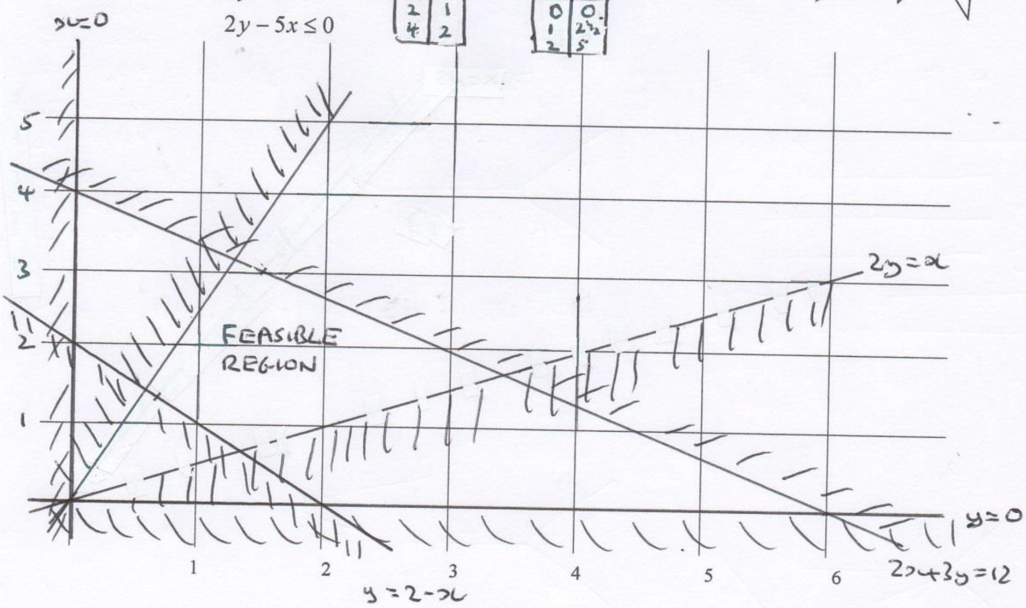
x	y
0	2
1	1
2	0

$2y=x$

x	y
0	0
2	1
4	2

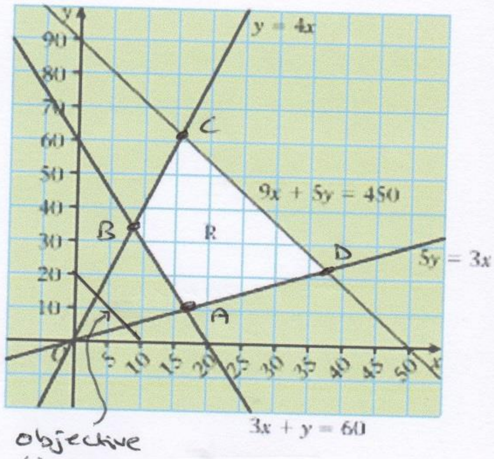
$2y-5x=0$

x	y
0	0
1	$2\frac{1}{2}$
2	5



Linear Programming

Find the optimal solution and optimal value using
a) Vertex method: maximise $J = x + 4y$
b) Ruler method: minimise $M = 2x + y$



objective line

Part a)

	Line Equations	Solutions of Simultaneous Equations	Value of J
A	$5y=3x$ $3x+y=60$	$x=50/3$ $y=10$	$\frac{170}{3}$ $=56.\dot{6}$
B	$y=4x$ $3x+y=60$	$x=60/7$ $y=240/7$	$\frac{1020}{7}$ $=145.71$
C	$y=4x$ $9x+5y=450$	$x=\frac{450}{29}$ $y=\frac{1800}{29}$	$\frac{7650}{29}$ $=263.79$
D	$5y=3x$ $9x+5y=450$	$x=75/2$ $y=45/2$	$\frac{255}{2}$ $=127.5$

Therefore the maximum is 263.79 which is achieved using $x=\frac{450}{29}$, $y=\frac{1800}{29}$

Part b) $2x+y=20$ shown on diagram

$\therefore$ minimum occurs at A, i.e. $x=\frac{50}{3}$, $y=10$

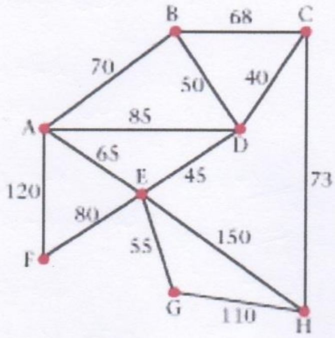
Route Inspection:

The network below represents the streets in a village. The number on each arc represents the length of the street in metres. The junctions have been labelled A, B, C, D, E, F, G, H. A salesman visits each house. He needs to travel along each street at least once. The total length of the streets is 1011m.

a) He parks his car at A and starts and finishes there. He wishes to minimise the distance he walks. Apply the route inspection algorithm and hence find a route he could take, stating the distance he would need to walk.

b) A friend offers to drive the salesman to a junction at the start of the day and collect him from another junction later in the day. Where should the salesman asked to be dropped off/picked up from and how far would he need to walk?

c) The friend now says he needs to drop the salesman off at B. Where should he be dropped off and how far would he need to walk?



Odd Nodes: B, C, E, H

Table of potential repeats: BC+EH=68+150 =218
BE+CH=95+73 =168*
Full Route Inspection BH+CE=141+85 =226

* Smallest is: 168 Hence repeat: BD, DE, CH

New total: 1011+ 168 = 1179

Route: A BDAEDEGH CBDCHEFA

Start/Finish in different places

Smallest individual entry is: 68 Hence repeat: BC

Start/Finish at: E, H

New total: 1011+68 = 1079

Route: EFAEGHEDABDCBCH

Start at B

Smallest individual entry not involving B is: CH=73

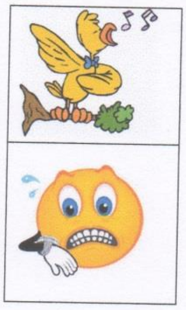
Hence repeat: CH

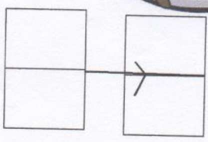
Start/Finish at: B, E

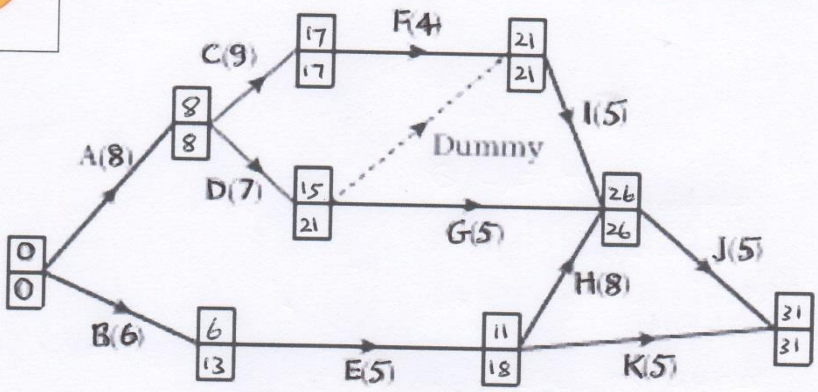
New total: 1011+73=1084

Route: B DABCHCDEAFEGHE

Critical Path Analysis:







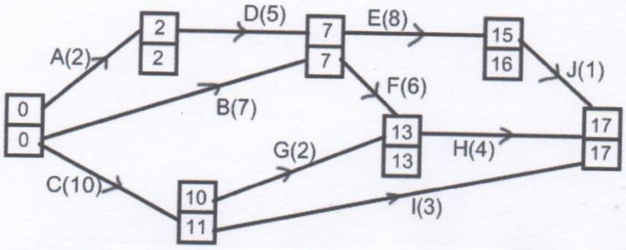
Critical Path:
= Longest path in network A C F I J
= Path of critical activities
= Path of activities with no float
= Path of activities which cannot be delayed without affecting the duration of the whole project.

Draw the dummy for each situation:

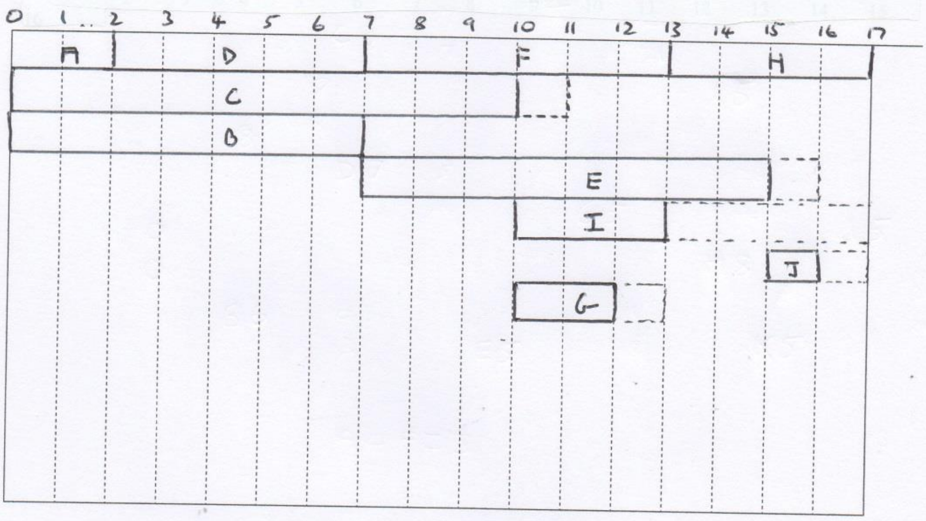
C and D depend on A and B
but E depends only on A

To enable unique representation
of J and K in terms of their end events

Gantt Charts, Scheduling

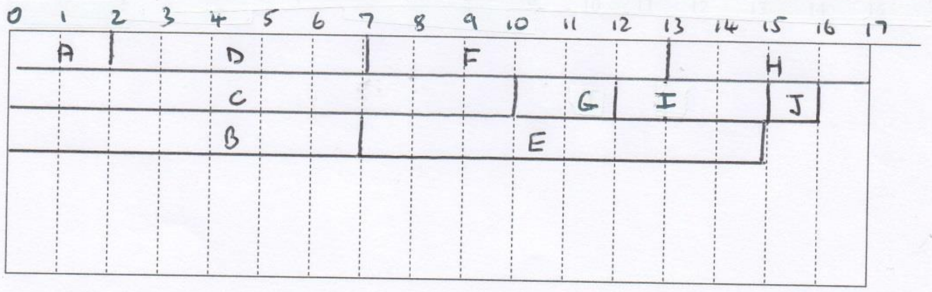


Draw a Gantt chart to represent this activity network.

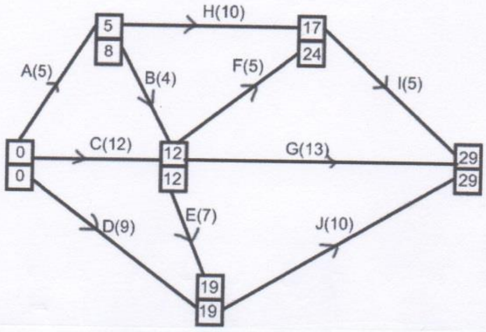


Which activities may be happening on day 11? FCEIG
Which activities must be happening on day 11? FE
What is the minimum number of workers needed to complete the job in 17 days? Why?

Schedule the jobs onto 3 workers.



Scheduling

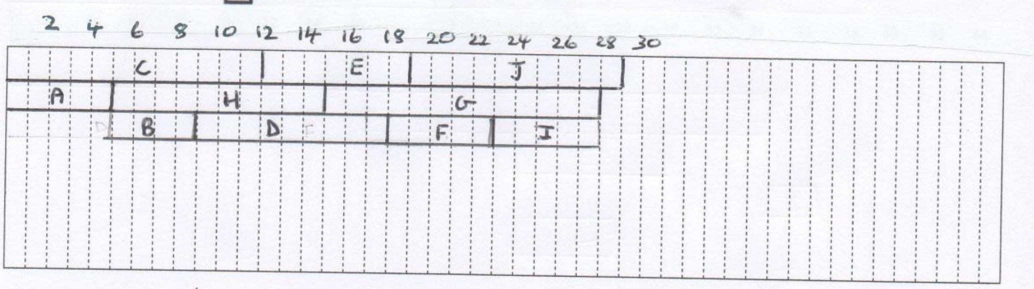


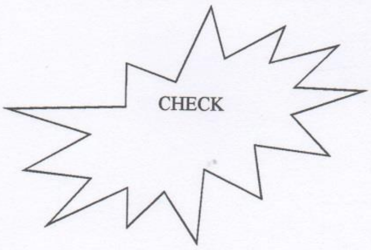
What is the lower bound of workers needed to complete the jobs in 29 days?

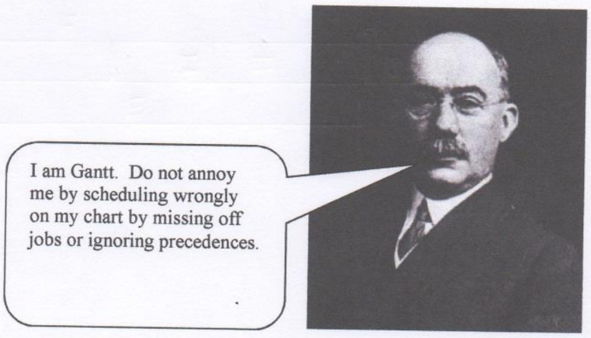
sum of activity times = $\frac{80}{29}$ = 2.76

∴ lower bound = 3

Schedule these jobs in the minimum time possible







Glossary for D1

Algorithms

In a list containing N items the 'middle' item has position $[\frac{1}{2}(N+1)]$ if N is odd, $[\frac{1}{2}(N+2)]$ if N is even, so that if $N = 9$, the middle item is the 5th and if $N = 6$ it is the 4th.

Algorithms on graphs

A **graph** G consists of points (**vertices** or **nodes**) which are connected by lines (**edges** or **arcs**).
A **subgraph** of G is a graph, each of whose vertices belongs to G and each of whose edges belongs to G.
If a graph has a number associated with each edge (usually called its **weight**) then the graph is called a **weighted graph** or **network**.
The **degree** or **valency** of a vertex is the number of edges incident to it. A vertex is **odd** **(even)** if it has **odd** **(even)** degree.
A **path** is a finite sequence of edges, such that the end vertex of one edge in the sequence is the start vertex of the next, and in which no vertex appears more than once.
A **cycle** **(circuit)** is a closed path, ie the end vertex of the last edge is the start vertex of the first edge.
Two vertices are **connected** if there is a path between them. A graph is **connected** if all its vertices are connected.
If the edges of a graph have a direction associated with them they are known as **directed edges** and the graph is known as a **digraph**.
A **tree** is a connected graph with no cycles.
A **spanning tree** of a graph G is a subgraph which includes all the vertices of G and is also a tree.
A **minimum spanning tree** (MST) is a spanning tree such that the total length of its arcs is as small as
possible. (MST is sometimes called a **minimum connector**.)
A graph in which each of the n vertices is connected to every other vertex is called a **complete graph**.

Critical path analysis

The **total float** $F(i, j)$ of activity (i, j) is defined to be $F(i, j) = l_j - e_i -$ duration (i, j), where e_i is the earliest time for event i and l_j is the latest time for event j.

Matchings

A **bipartite graph** consists of two sets of vertices X and Y. The edges only join vertices in X to vertices in Y, not vertices within a set. (If there are r vertices in X and s vertices in Y then this graph is $K_{r,s}$.)
A **matching** is the pairing of some or all of the elements of one set, X, with elements of a second set, Y. If every member of X is paired with a member of Y the matching is said to be a **complete matching**.