













## Mechanics

### Kinematics

$$v = u + at$$

$$s = ut + \frac{1}{2}at^2$$

$$s = \frac{1}{2}(u + v)t$$

$$v^2 = u^2 + 2as$$

$$s = vt - \frac{1}{2}at^2$$

### Newton's experimental law

Between two smooth spheres  $v_1 - v_2 = -e(u_1 - u_2)$

Between a smooth sphere with a fixed plane surface  $v = -eu$

### Motion in a circle

Tangential velocity is  $v = r\dot{\theta}$

Radial acceleration is  $\frac{v^2}{r}$  or  $r\dot{\theta}^2$  towards the centre

Tangential acceleration is  $\dot{v} = r\ddot{\theta}$

## Discrete

### Sorting algorithms

Bubble sort:

Start at the left hand end of the list unless specified otherwise.

Compare the first and second values and swap if necessary. Then compare the (new) second value with the third value and swap if necessary. Continue in this way until all values have been considered.

Fix the last value then repeat with the reduced list until either there is a pass in which no swaps occur or the list is reduced to length 1, then stop.

Shuttle sort:

Start at the left hand end of the list unless specified otherwise.

Compare the second value with the first and swap if necessary, this completes the first pass. Next compare the third value with the second and swap if necessary, if a swap happened shuttle back to compare the (new) second with the first as in the first pass, this completes the second pass.

Next compare the fourth value with the third and swap if necessary, if a swap happened shuttle back to compare the (new) third value with the second as in the second pass (so if a swap happens shuttle back again). Continue in this way for  $n-1$  passes, where  $n$  is the length of the list.

## Network algorithms

### Dijkstra's algorithm

START with a graph  $G$ . At each vertex draw a box, the lower area for temporary labels, the upper left hand area for the order of becoming permanent and the upper right hand area for the permanent label.

STEP 1 Make the given start vertex permanent by giving it permanent label 0 and order label 1.

STEP 2 For each vertex that is not permanent and is connected by an arc to the vertex that has just been made permanent (with permanent label =  $P$ ), add the arc weight to  $P$ . If this is smaller than the best temporary label at the vertex, write this value as the new best temporary label.

STEP 3 Choose the vertex that is not yet permanent which has the smallest best temporary label. If there is more than one such vertex, choose any one of them. Make this vertex permanent and assign it the next order label.

STEP 4 If every vertex is now permanent, or if the target vertex is permanent, use 'trace back' to find the routes or route, then STOP; otherwise return to STEP 2.

### Prim's algorithm (graphical version)

START with an arbitrary vertex of  $G$ .

STEP 1 Add an edge of minimum weight joining a vertex already included to a vertex not already included.

STEP 2 If a spanning tree is obtained STOP; otherwise return to STEP 1.

### Prim's algorithm (tabular version)

START with a table (or matrix) of weights for a connected weighted graph.

STEP 1 Cross through the entries in an arbitrary row, and mark the corresponding column.

STEP 2 Choose a minimum entry from the uncircled entries in the marked column(s).

STEP 3 If no such entry exists STOP; otherwise go to STEP 4.

STEP 4 Circle the weight  $w_{ij}$  found in STEP 2; mark column  $j$ ; cross through row  $i$ .

STEP 5 Return to STEP 2.

### Kruskal's algorithm

START with all the vertices of  $G$ , but no edges; list the edges in increasing order of weight.

STEP 1 Add an edge of  $G$  of minimum weight in such a way that no cycles are created.

STEP 2 If a spanning tree is obtained STOP; otherwise return to STEP 1.

## Additional Pure

### Vector product

$\mathbf{a} \times \mathbf{b} = |\mathbf{a}||\mathbf{b}|\sin \theta \hat{\mathbf{n}}$ , where  $\mathbf{a}, \mathbf{b}, \hat{\mathbf{n}}$ , in that order form a right-handed triple.