

1 Task 1

```
1 function plott = task1(steps, start, left_prob, no_step_prob)
2     num = steps; %number of steps
3     sum = start; %starting position of the person
4     plott(1) = start; %the steps-position plot
5     step_direction = rand(num, 1); %random number which decides his movement
6
7     for i = 2:num
8         if step_direction(i) < no_step_prob
9             sum = sum;
10        elseif step_direction(i) < no_step_prob + left_prob
11            sum = sum - 1;
12        else
13            sum = sum + 1;
14        end
15        plott(i) = sum;
16    end
17
18    %figure (1);
19    %c = plot(1:num, plott, '-rx');
20    %set(c, 'color', 'blue');
21    %grid on;
22    %title('random walk in 1 dimension')
23    %xlabel('steps');
24 end
```

```

1 function avgstep = task1Avg(start, steps, left_prob, no_step_prob, simulations, step_needed)
2     avgplott(1) = start;
3     all_simulations = zeros(simulations, steps);
4     for i = 1:simulations
5         all_simulations(i,:) = task1(steps, start, left_prob, no_step_prob);
6     end
7
8     for j = 2:steps
9         sum = 0;
10        for i = 1:simulations
11            sum = sum + all_simulations(i, j);
12        end
13        avgplott(j) = sum/simulations;
14    end
15    avgstep = avgplott(step_needed);
16
17    figure (1);
18    c = plot(1:steps, avgplott, '-rx');
19    set(c, 'color', 'blue');
20    grid on;
21    title('avg random walk in 1 dimension')
22    ylabel('steps');
23 end

```

2 Task 2

```
1 function task2(steps, distance_apart, left_prob1, left_prob2, no_step_prob1, no_step_prob2)
2     start1 = 0;
3     start2 = 0 + distance_apart;
4     sum1 = start1;
5     sum2 = start2;
6     num = steps;
7     step_direction1 = rand(num, 1);
8     step_direction2 = rand(num, 1);
9
10    plott_1(1) = start1;
11    plott_2(1) = start2;
12
13    for i = 2:num
14        if step_direction1(i) < no_step_prob1
15            sum1 = sum1;
16        elseif step_direction1(i) < no_step_prob1 + left_prob1
17            sum1 = sum1 - 1;
18        else
19            sum1 = sum1 + 1;
20        end
21        plott_1(i) = sum1;
22
23        if step_direction2(i) < no_step_prob2
24            sum2 = sum2;
25        elseif step_direction2(i) < no_step_prob2 + left_prob2
26            sum2 = sum2 - 1;
27        else
28            sum2 = sum2 + 1;
29        end
30        plott_2(i) = sum2;
31    end
```

```
32
33     figure (1);
34     c = plot(1:num, plott_1, '-rx');
35     hold on;
36     b = plot(1:num, plott_2, '-rx');
37     set(c, 'color', 'red');
38     set(b, 'color', 'blue');
39     grid on;
40     title('Two People doing random walk in 1 dimension')
41     xlabel('steps');
42     hold off;
43 end
```

```

1 function avgmeeting_step = task2Avg(start_1, distance, steps, left_prob, no_step_prob, simulations)
2     avgplott_1(1) = start_1;
3     avgplott_2(1) = start_1 + distance;
4     all_simulations_1 = zeros(simulations, steps);
5     all_simulations_2 = zeros(simulations, steps);
6     for i = 1:simulations
7         all_simulations_1(i,:) = task1(steps, start_1, left_prob, no_step_prob);
8         all_simulations_2(i,:) = task1(steps, start_1 + distance, left_prob, no_step_prob);
9     end
10
11     for j = 2:steps
12         sum_1 = 0;
13         sum_2 = 0;
14         for i = 1:simulations
15             sum_1 = sum_1 + all_simulations_1(i, j);
16             sum_2 = sum_2 + all_simulations_2(i, j);
17         end
18         avgplott_1(j) = sum_1/simulations;
19         avgplott_2(j) = sum_2/simulations;
20     end
21
22     avgmeeting_plott = zeros(1,simulations);
23     for i = 1:simulations
24         avgmeeting_plott(1, i) = -1;
25         for j = 1:steps
26             if all_simulations_2(i, j) - all_simulations_1(i, j) <= 0
27                 avgmeeting_plott(1, i) = j;
28                 break
29             end
30         end
31     end
32
33 %     sum_meeting_step = 0;
34 %     for i = 1:simulations

```

```

35 %         sum_meeting_step = sum_meeting_step + avgmeeting_plott(1, i);
36 %     end
37 %
38 %     mean_sum = mean(avgmeeting_plott,'all');
39 %     disp(mean_sum)
40
41 %     avgmeeting_step = sum_meeting_step/simulations;
42
43     avgmeeting_step = mean(avgmeeting_plott,'all');
44
45     figure (1);
46     c = plot(1:steps, avgplott_1, '-rx');
47     hold on;
48     b = plot(1:steps, avgplott_2, '-rx');
49     set(c, 'color', 'red');
50     set(b, 'color', 'blue');
51     grid on;
52     title('two people doing random walk in 1 dimension')
53     xlabel('steps');
54     hold off;
55 end

```

3 Task 3

```
1 function task3(steps, startx, starty, prob_nostep, prob_halfstep, prob_left, prob_right, prob_down)
2     num = steps;
3     sumx = startx;
4     sumy = starty;
5     plottx(1) = startx;
6     plotty(1) = starty;
7     %step_direction = randi([1,4],1,num);
8     step_direction = rand(num, 1);
9     step_size = rand(num, 1);
10    step = 0;
11
12    for i = 2:num
13        if step_size(i) < prob_nostep
14            step = 0;
15        elseif step_size(i) < prob_nostep + prob_halfstep
16            step = 0.5;
17        else
18            step = 1;
19        end
20        if step_direction(i) < prob_left
21            sumx = sumx - step;
22            if ((sumx-startx)^2 + (sumy-starty)^2)^(1/2) > 100
23                sumx = sumx + 2*step;
24            end
25        elseif step_direction(i) < prob_left + prob_right
26            sumx = sumx + step;
27            if ((sumx-startx)^2 + (sumy-starty)^2)^(1/2) > 100
28                sumx = sumx - 2*step;
29            end
30        elseif step_direction(i) < prob_left + prob_right + prob_down
31            sumy = sumy - step;
```

```

32         if ((sumx-startx)^2 + (sumy-starty)^2)^(1/2) > 100
33             sumy = sumy + 2*step;
34         end
35     else
36         sumy = sumy + step;
37         if ((sumx-startx)^2 + (sumy-starty)^2)^(1/2) > 100
38             sumy = sumy - 2*step;
39         end
40     end
41     plottx(i) = sumx;
42     plotty(i) = sumy;
43 end
44
45 figure (1);
46 c = plot3(plottx, plotty, 1:num, '-rx');
47 set(c, 'color', 'red');
48 grid on;
49 numm = num2str(num);
50 str1 = 'Discrete Random Walk in 2 dimension (n = ';
51 str2 = ' steps)';
52 result = [str1 numm str2];
53 xlabel('x-axis');
54 ylabel('y-axis');
55 zlabel('time');
56 title(result);
57 end

```


4 Task 4

```
1 function plott = task4(steps, start, left_prob, no_step_prob)
2     num = steps; %number of steps
3     sum = start; %starting position of the person
4     plott(1) = start; %the steps-position plot
5     step_direction = rand(num, 1); %random number which decides his movement
6     step_size = rand(num, 1);
7
8     for i = 2:num
9         if step_direction(i) < no_step_prob
10             sum = sum;
11         elseif step_direction(i) < no_step_prob + left_prob
12             sum = sum - step_size(i);
13         else
14             sum = sum + step_size(i);
15         end
16         plott(i) = sum;
17     end
18
19     figure (1);
20     c = plot(plott, 1:num, '-rx');
21     set(c, 'color', 'blue');
22     grid on;
23     title('random walk in 1 dimension')
24     ylabel('steps');
25 end
```

```

1 function avgstep = task4Avg(start, steps, left_prob, no_step_prob, simulations, step_needed)
2     avgplott(1) = start;
3     all_simulations = zeros(simulations, steps);
4     for i = 1:simulations
5         all_simulations(i,:) = task4(steps, start, left_prob, no_step_prob);
6     end
7
8     for j = 2:steps
9         sum = 0;
10        for i = 1:simulations
11            sum = sum + all_simulations(i, j);
12        end
13        avgplott(j) = sum/simulations;
14    end
15    avgstep = avgplott(step_needed);
16
17    figure (1);
18    c = plot(1:steps, avgplott, '-rx');
19    set(c, 'color', 'blue');
20    grid on;
21    title('avg random walk in 1 dimension')
22    ylabel('steps');
23 end

```

5 Task 5

```
1 function task5(steps, startx, starty)
2     num = steps;
3     sumx = startx;
4     sumy = starty;
5     plottx(1) = startx;
6     plotty(1) = starty;
7     theta = 2*pi*rand(num,1);
8     r = rand(num, 1);
9
10    for i = 2:num
11        sumx = sumx + r(i)*cos(theta(i));
12        if ((sumx-startx)^2 + (sumy-starty)^2)^(1/2) > 100
13            sumx = sumx - 2*(r(i)*cos(theta(i)));
14        end
15        sumy = sumy + r(i)*sin(theta(i));
16        if ((sumx-startx)^2 + (sumy-starty)^2)^(1/2) > 100
17            sumy = sumy - 2*(r(i)*sin(theta(i)));
18        end
19        plottx(i) = sumx;
20        plotty(i) = sumy;
21    end
22
23    figure (1);
24    c = plot3(plottx, plotty, 1:num, '-rx');
25    set(c, 'color', 'red');
26    grid on;
27    numm = num2str(num);
28    str1 = 'person doing randomwalk in 2 dimension (n = ';
29    str2 = ' steps)';
30    result = [str1 numm str2];
31    xlabel('x-axis');
```

```
32     ylabel('y-axis');
33     zlabel('time');
34     title(result);
35 end
```

```

1 function plottxy = task5helper(steps, startx, starty)
2     num = steps;
3     sumx = startx;
4     sumy = starty;
5     plottx(1) = startx;
6     plotty(1) = starty;
7     theta = 2*pi*rand(num,1);
8     r = rand(num, 1);
9
10    for i = 2:num
11        sumx = sumx + r(i)*cos(theta(i));
12        if ((sumx-startx)^2 + (sumy-starty)^2)^(1/2) > 100
13            sumx = sumx - 2*(r(i)*cos(theta(i)));
14        end
15        sumy = sumy + r(i)*sin(theta(i));
16        if ((sumx-startx)^2 + (sumy-starty)^2)^(1/2) > 100
17            sumy = sumy - 2*(r(i)*sin(theta(i)));
18        end
19        plottx(i) = sumx;
20        plotty(i) = sumy;
21    end
22
23    % figure (1);
24    % c = plot(plottx, plotty, '-rx', 'Color', [rand, rand, ran]);
25    %
26    % grid on;
27    % numm = num2str(num);
28    % str1 = 'person doing randomwalk in 2 dimension (n = ';
29    % str2 = ' steps)';
30    % result = [str1 numm str2];
31    % xlabel('x-axis');
32    % ylabel('y-axis');
33    % zlabel('time');
34    % title(result);

```

```
35 %  
36     plotxy = zeros(2, steps);  
37     plottxy(1, :) = plottx;  
38     plottxy(2, :) = plotty;  
39  
40 end
```

6 Task 7

```
1 function task7(steps, startx, starty, prob_nostep, prob_halfstep)
2     num = steps;
3     sumx = startx;
4     sumy = starty;
5     plottx(1) = startx;
6     plotty(1) = starty;
7     theta = 2*pi*rand(num,1);
8     step_size = rand(num, 1);
9
10    for i = 2:num
11        if step_size(i) < prob_nostep
12            r = 0;
13        elseif step_size(i) < prob_nostep + prob_halfstep
14            r = 0.5;
15        else
16            r = 1;
17        end
18        sumx = sumx + r*cos(theta(i));
19        if ((sumx-startx)^2 + (sumy-starty)^2)^(1/2) > 100
20            sumx = sumx - 2*(r*cos(theta(i)));
21        end
22        sumy = sumy + r*sin(theta(i));
23        if ((sumx-startx)^2 + (sumy-starty)^2)^(1/2) > 100
24            sumy = sumy - 2*(r*sin(theta(i)));
25        end
26        plottx(i) = sumx;
27        plotty(i) = sumy;
28    end
29
30    figure (1);
31    c = plot3(plottx, plotty, 1:num, '-rx');
```

```

32     set(c, 'color', 'red');
33     grid on;
34     numm = num2str(num);
35     str1 = 'person doing randomwalk in 2 dimension (n = ';
36     str2 = ' steps)';
37     result = [str1 numm str2];
38     xlabel('x-axis');
39     ylabel('y-axis');
40     zlabel('time');
41     title(result);
42 end

```

7 Task 8

```

1
2 function avg_step = task8(sim, steps, a_x, a_y, b_x, b_y)
3
4     nums = sim;
5     steps = steps;
6     particles1 = zeros(2, steps, nums);
7     particles2 = zeros(2, steps, nums);
8
9     for i = 1:nums
10         particle1(:, :, i) = task5helper(steps, a_x, a_y);
11         particle2(:, :, i) = task5helper(steps, b_x, b_y);
12     end
13
14     step_arr = zeros(1, steps);
15     % mean_step_arr = zeros(1, nums);
16
17
18     for i = 1:nums

```



```
19         for j = 1:steps
20             if ((particle2(1,j,i)-particle1(1,j,i))^2 +(particle2(2,j,i)-particle1(2,j,i)^2))^(1/2) <= 1
21                 step_arr(i) = j;
22             end
23         end
24     end
25 end
26
27
28 avg_step = mean(step_arr, 'all');
29
30 end
```

```

1 function task8sim(sim,sim_internal, steps,a_x, a_y, b_x, b_y)
2
3     num = sim;
4     mean_step_arr = zeros(1, num);
5     for i = 1:num
6         mean_step_arr(i) = task8(sim_internal, steps, a_x, a_y, b_x, b_y);
7     end
8
9     mean_mean = mean(mean_step_arr, 'all');
10    figure(1);
11    c = plot(1:num, mean_step_arr, 'r');
12
13    euc_dist = ((a_x-b_x)^2+(a_y-b_y)^2)^(1/2);
14    euc_dist = num2str(euc_dist);
15
16    xlabel('Simulation Number');
17    ylabel('Mean of Step at which 2 Particles are within 1 unit of distance');
18    mean_mean = num2str(mean_mean);
19    str1 = 'Expected Steps of Proximity When Starting from a Distance of ';
20    str2 = ' with a deviation of ';
21    result = [str1 euc_dist str2 mean_mean];
22    title(result);
23
24 end

```

8 Task 9

```
1 function avg_step = task93(sim, steps, a_x, a_y, b_x, b_y, c_x, c_y)
2
3     nums = sim;
4     steps = steps;
5     particles1 = zeros(2, steps, nums);
6     particles2 = zeros(2, steps, nums);
7     particles3 = zeros(2, steps, nums);
8
9     for i = 1:nums
10         particle1(:, :, i) = task5helper(steps, a_x, a_y);
11         particle2(:, :, i) = task5helper(steps, b_x, b_y);
12         particle3(:, :, i) = task5helper(steps, c_x, c_y);
13     end
14
15     step_arr = zeros(1, steps);
16     % mean_step_arr = zeros(1, nums);
17
18
19     for i = 1:nums
20         for j = 1:steps
21             a = [particle1(1,j,i), particle1(2,j,i)];
22             b = [particle2(1,j,i), particle2(2,j,i)];
23             c = [particle3(1,j,i), particle3(2,j,i)];
24             if norm(a-b) <= 6 && norm(a-c) <=6 && norm(c-b) <=6
25                 step_arr(i) = j;
26             end
27         end
28     end
29 end
30
31
```

```
32 avg_step = mean(step_arr, 'all');  
33  
34 end
```

```

1 function task93sim(sim,sim_internal, steps,a_x, a_y, b_x, b_y, c_x, c_y)
2
3     num = sim;
4     mean_step_arr = zeros(1, num);
5     for i = 1:num
6         mean_step_arr(i) = task93(sim_internal, steps, a_x, a_y, b_x, b_y, c_x, c_y);
7     end
8
9     mean_mean = mean(mean_step_arr, 'all');
10    figure(1);
11    c = plot(1:num, mean_step_arr, 'r');
12
13    euc_dist = ((a_x-b_x)^2+(a_y-b_y)^2)^(1/2);
14    euc_dist = num2str(euc_dist);
15
16    xlabel('Simulation Number');
17    ylabel('Mean of Step at which 3 Particles are within 1 unit of distance');
18    mean_mean = num2str(mean_mean);
19    str1 = 'Expected Steps of Proximity When Starting from a Distance of ';
20    str2 = ' with a deviation of ';
21    result = [str1 euc_dist str2 mean_mean];
22    title(result);
23
24 end

```

```

1 function avg_step = task94(sim, steps, a_x, a_y, b_x, b_y, c_x, c_y, d_x, d_y)
2
3     nums = sim;
4     steps = steps;
5     particles1 = zeros(2, steps, nums);
6     particles2 = zeros(2, steps, nums);
7     particles3 = zeros(2, steps, nums);
8     particles4 = zeros(2, steps, nums);
9
10    for i = 1:nums
11        particle1(:, :, i) = task58(steps, a_x, a_y);
12        particle2(:, :, i) = task58(steps, b_x, b_y);
13        particle3(:, :, i) = task58(steps, c_x, c_y);
14        particle4(:, :, i) = task58(steps, d_x, d_y);
15    end
16
17    step_arr = zeros(1, steps);
18    % mean_step_arr = zeros(1, nums);
19
20
21    for i = 1:nums
22        for j = 1:steps
23            a = [particles1(1,j,i), particles1(2,j,i)];
24            b = [particles2(1,j,i), particles2(2,j,i)];
25            c = [particles3(1,j,i), particles3(1,j,i)];
26            d = [particles4(1,j,i), particles4(1,j,i)];
27
28            if norm(a-b) <= 6 && norm(a-c) <=6 && norm(a-d) <=6 && norm(b-c) <=6 && norm(b-d) <= 6 && norm(c-d) <=6
29
30                step_arr(i) = j;
31
32            end
33        end
34    end

```

```
35
36
37 avg_step = mean(step_arr, 'all');
38
39 end
```

```

1 function task94sim(sim,sim_internal, steps,a_x, a_y, b_x, b_y, c_x, c_y, d_x, d_y)
2
3     num = sim;
4     mean_step_arr = zeros(1, num);
5     for i = 1:num
6         mean_step_arr(i) = task94(sim_internal, steps, a_x, a_y, b_x, b_y, c_x, c_y, d_x, d_y);
7     end
8
9     mean_mean = mean(mean_step_arr, 'all');
10    figure(1);
11    c = plot(1:num, mean_step_arr, 'r');
12
13
14
15    xlabel('Simulation Number');
16    ylabel('Mean of Step at which 3 Particles are within 1 unit of distance');
17    mean_mean = num2str(mean_mean);
18    str1 = 'Expected Steps of Proximity When Starting from a Distance of ';
19    str2 = ' with a deviation of ';
20    result = [str1 euc_dist str2 mean_mean];
21    title(result);
22
23 end

```