

Problem 1.18.9. Write a class called **Snake and Ladder** as explained in figure 1.24. The expected output is shown in the figure 1.25 and 1.26

Snake and the ladder game

```

int[][] l = {
    {29, 38},
    {4, 14},
    {9, 31},
    {21, 42},
    {28, 84},
    {36, 44},
    {51, 67},
    {71, 91},
    {80, 100},
};

int[][] s = {
    {98, 78},
    {95, 75},
    {93, 73},
    {87, 24},
    {64, 60},
    {62, 19},
    {56, 53},
    {49, 11},
    {16, 6},
    {47, 26},
};
    
```

one("Board 100",100,l,s,500000) ; //Try first for 10

abstract void play() ;
MUST WRITE CODE IN SnakeAndLadder.java
 Cannot change anything in SnakeAndLadderBase.java
 Upload: SnakeAndLadder.java and the output of the program

Figure 1.24: Snake and problem gate

1.18. PROBLEM SET

```
one("Board 100",100,l,s,10) ;  
  
SnakeAndLadder problem STARTS  
1->6->7->(9)31->32->(36)44->46->50->(51)67-> DELETED->100 Num dice rolled = 43  
1->7->12->15->18->23->26->(28)84-> DELETED -L96->99->99->99->100 Num dice rolled = 17  
1->6->10->14->(16)6->(9)31->33->39->41-DELETED->78->(80)100 Num dice rolled = 21  
  
DELETED  
  
1->6->8->DELETED27->91->96->96->99->99->99->99->99->99->99->99->99->100 Num dice rolled = 82  
-----Board 100 -----  
Number of squares = 100  
For 10 rounds: min = 17 max = 82  
-----Min game -----  
1->7->12->15->18->23->26->(28)84->85->91->92->96->96->96->99->99->99->100  
-----max game -----  
1->6->8->13->18->23->27->1->(16)6->7->11  
->91->96->96->DELETED---99->99->99->99->99->99->99->99->99->100  
Run time = 0.008689500000000001 secs  
SnakeAndLadder problem ENDS
```

Figure 1.25: Expected output when number of rounds is < 20

SnakeAndLadder problem STARTS

-----Board 100 -----

Number of squares = 100

For 500000 rounds: min = 7 max = 373

-----Min game -----

1->(4)14->19->22->(28)84->88->94->100

-----max game -----

1->3->5->7->(9)31->34->39->42->48->54->59->65->67->
 73->77->79->82->88->92->(95)75->77->78->81->85->90->
 (93)73->76->78->82->88->92->(93)73->78->79->83->(87)24->
 27->32->(36)44->(49)11->17->(21)42->46->(47)26->31->32->
 37->41->42->44->(47)26->30->34->40->46->50->(56)53->58->59->
 60->65->67->68->72->77->81->84->86->89->(93)73->79->84->
 (87)24->30->35->37->42->44->46->50->52->54->55->58->(64)60->
 66->68->74->76->79->84->(87)24->(29)38->44->(47)26->31->35->
 38->39->45->(47)26->(29)38->44->45->(49)11->13->15->20->
 (21)42->43->(49)11->17->23->26->32->35->(36)44->46->50->
 55->59->60->(62)19->25->27->30->32->38->40->42->43->45->
 50->(56)53->54->57->58->59->(64)60->(62)19->24->30->35->39->
 44->(47)26->(29)38->42->46->(47)26->32->(36)44->(49)11->
 17->(21)42->(47)26->27->31->35->40->44->(47)26->30->(36)44->
 45->46->48->52->58->(62)19->20->22->26->(29)38->39->42->
 43->45->48->50->53->55->57->60->66->67->68->70->72->75->
 76->81->(87)24->(28)84->88->90->(95)75->81->83->(87)24->26->
 27->31->35->41->43->45->48->(49)11->15->18->24->(29)38->40->
 43->(49)11->12->18->23->(29)38->39->40->41->42->46->(49)11->
 13->19->20->26->27->30->31->32->35->41->(47)26->32->
 (36)44->45->46->(49)11->15->20->(21)42->46->(51)67->69->(71)91->
 (93)73->77->81->(87)24->25->(29)38->42->48->54->55->58->60->
 65->69->73->78->79->83->84->85->88->92->(93)73->78->84->90->
 (95)75->78->83->(87)24->(28)84->85->(87)24->27->(29)38->43->45->
 (51)67->68->74->79->81->(87)24->(29)38->43->46->52->57->58->
 60->(62)19->25->30->32->35->37->39->43->44->50->(51)67->69->
 73->78->82->86->(87)24->(28)84->85->(87)24->27->30->(36)44->
 46->(47)26->30->33->37->42->45->(49)11->15->(16)6->12->17->(21)42->44->
 46->(49)11->(16)6->10->13->18->23->27->(29)38->43->45->46->(47)26->
 (29)38->42->44->46->(51)67->70->76->82->86->91->94->(95)75->77->83->
 89->92->94->(98)78->81->83->85->91->(93)73->79->(80)100

Run time = 0.8624951999999999 secs For 500000 rounds

SnakeAndLadder problem ENDS

CAN YOU BEAT ME?

Figure 1.26: Expected output when number of rounds is > 20