# Snake and Ladders : The Quickest Way Up

Crack-A-Hack

01FE15BEC162
Sanath
Vernekar

# Problem statement:

Markov takes out his Snakes and Ladders game and stares at the board, and wonders: If he had absolute control on the die (singular), and could get it to generate any number (in the range 1-6) he desired, what would be the least number of rolls of the die in which he'd be able to reach the destination square (Square Number 100 ) after having started at the base square (Square Number 1)?

# Code in C++:-

```
#include <iostream>
#include <vector>
#include <queue>
using namespace std;
const int N=104;
const int INF=100000000;
int main()
{
	int t;
	cin>>t;
	for(int k=1;k<=t;++k)
	{
		vector<int> graph(N,0);
		vector<bool> mark(N,false);
		int n,m;
		cin>>n;
		for(int i=0;i<n;++i)
		{
			int a,b;
			cin>>a>>b;
			graph[a]=b;
		}
		cin>>n;
		for(int i=0;i<n;++i)
		{
			int a,b;
			cin>>a>>b;
			graph[a]=b;
		}
		queue< pair<int,int> > q;
		int ans=INF;
		q.push(make_pair(1,0));
		mark[1]=true;
		while(!q.empty())
```

```
		{
			pair<int,int> p=q.front();
			if(p.first==100)
			{
				ans=p.second;
				break;
			}
			q.pop();
			for(int i=1;i<=6;++i)
			{
				int x=p.first+i;
				if(x>100)
					continue;
				if(mark[x]==false)
				{
					mark[x]=true;
					if(graph[x]==0)
						q.push(make_pair(x,p.second+1));
					else
					{
						x=graph[x];
						mark[x]=true;
						q.push(make_pair(x,p.second+1));
					}
				}
			}
		}
		if(ans==INF)
			cout<<-1<<endl;
		else
			cout<<ans<<endl;

	}
}
```

Reference :
https://www.hackerrank.com/challenges/the-quickest-way-up/problem
https://www.geeksforgeeks.org/snake-ladder-problem-2