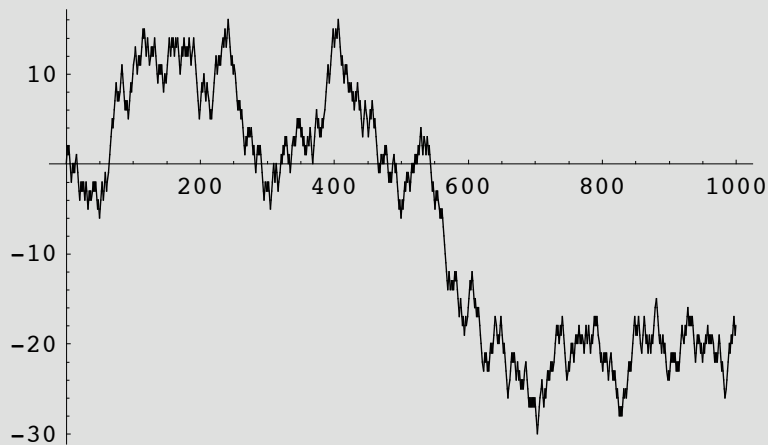


## Worksheet zu Gambler's ruin

### Symmetrischer random walk (Wahrscheinlichkeit $1/2$ für *up* und für *down*)

```
walk[n_] :=  
Module[{S, T},  
  T = Table[Random[Integer, {0, 1}], {i, 1, n}] /. 0 -> -1;  
  S = Table[{k, Sum[T[[j]], {j, 1, k}]}, {k, 0, n}];  
  ListPlot[S, PlotJoined -> True]  
]
```

walk[1000]

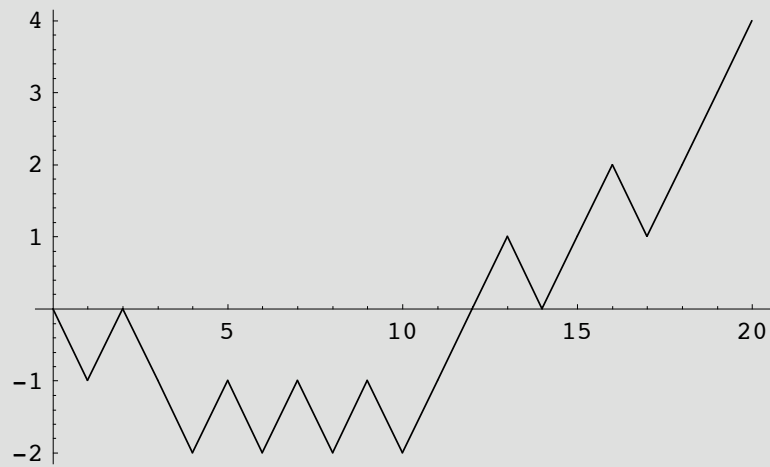


-Graphics-

### Random walk mit bias (Wahrscheinlichkeit $p$ für *up* und $1 - p$ für *down*)

```
biasedwalk[n_, p_] :=  
Module[{S, T},  
  T = Map[Function[x, Sign[p - x]],  
    Table[Random[Real, {0, 1}, 3], {i, 1, n}]];  
  S = Table[{k, Sum[T[[j]], {j, 1, k}]}, {k, 0, n}];  
  ListPlot[S, PlotJoined -> True]  
]
```

```
biasedwalk[20, 0.6]
```



- Graphics -

### Gambler's ruin

unteres Niveau ("Ruin") = **low**

oberes Niveau ("Gewinn") = **high**

Bias mit up – Wahrscheinlichkeit = **prob**

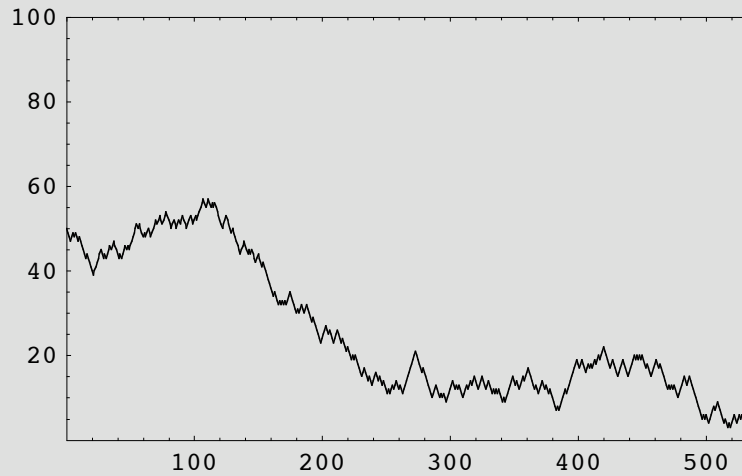
Startkapital = **offset**

Zeitbeschränkung = **bound**

```
sumit[a_, low_, high_, offset_] :=
Module[{l, s, i, b},
  ll = Length[a];
  s = offset;
  b = Append[{}, offset];
  i = 1;
  While[(s > low && s < high && i < ll),
    {s = s + a[[i]]; b = Append[b, s]; i++};
  b
]
```

```
gambler[low_, high_, start_, prob_, bound_] :=  
Module[{},  
  If[prob ≤ 0 || prob ≥ 1, Error];  
  If[start ≤ low || start ≥ high, Error];  
  S = Map[Function[x, Sign[prob - x]],  
    Table[Random[Real, {0, 1}, 3], {i, 1, bound}]];  
  T = sumit[S, low, high, start];  
  t = Length[T];  
  U = Table[{k - 1, T[[k]]}, {k, 1, t}];  
  ListPlot[U, PlotJoined → True,  
    PlotRange → {{0, Length[T]}, {low, high}}, Frame → True]  
]
```

```
gambler[0, 100, 50, 0.45, 1000]
```



- Graphics -