1 Problem Statement

We have children in a circle with their teacher. The teacher begins with a potato in hand. The children want to play the hot potato game. They are given a potato and can pass the potato to the person on their right or left decided by a fair coin. The game is won by the last child who does not touch the potato. (This is from the fivethirtyeight Riddler August 4, 2017 puzzle)

2 Answer

The answer can be arrived at via simulation most easily. It can be translated into a random walk problem, if an exact answer is desired. That solution gives 1/30 as the probabilitiy for each child. That is, it is equiprobable for each child to win. You might think that the children nearest the teacher have a disadvantage but this actually isn't so because no direction is preferred, and so it is as likely to go away from a beginning child as immediately eliminating the child.

What is more interesting is if you do not use a fair coin. If you change from a 50% chance of going to either direction to favoring a direction by as little as 1%, you find that you want to be on the side that is not favored by your "biased coin".

hotpotato.py

```python
#!/usr/bin/env python3

import numpy as np
import matplotlib.pyplot as plt

# inspired by Riddler Aug. 4 2017 puzzle

#children in a circle passing a hot potato

# make reproducible with random seed
np.random.seed(2)

# returns index of child to get potato
def indexmaker(index,children):
  # if index negative go to last
  # entry/child
  if index<0:
    return children+(index)
  # if index larger than
  # number of children
  # go to first entry/child
  elif index>=children:
    return index-children
  # otherwise do nothing
  else:
    return index


# probability of passing left and right are related
# go through game to determine winner.
def trial(childrow,childnum,leftprob=0.5,potatoindex=0):
  while np.sum(childrow)<(childnum-1):
    test=np.random.random(1)[0]
    if (test<leftprob):
      potatoindex=indexmaker(potatoindex-1,childnum)
    else:
      potatoindex=indexmaker(potatoindex+1,childnum)
    childrow[potatoindex]=1
  childwinner=np.argmin(childrow)
  return childwinner

# childnum = children + 1 teacher
```

```
childnum=31
#childnum=5
# winnerrow is a row with number of wins
# for each child
winnerrow=np.zeros(childnum)

# further generalizations possible
rightprob=0.50
leftprob=1-rightprob

# number of games to play
tottrials=int(1e4)
for i in range(tottrials):
  # set up each game each time
  childrow=np.zeros(childnum)
  childrow[0]=1
  potatoindex=0
  # determine winner
  winner=trial(childrow,childnum,leftprob,potatoindex)
  # add winner to winnerrow table to save
  winnerrow[winner]+=1
# print wins for each child
print(winnerrow)
# print probability of win for each child
print(1.*winnerrow/tottrials)
```