

# DESIGN AND ANALYSIS OF ALGORITHMS

## LAB 2 18<sup>th</sup> January, 2017

### C Code

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>

#define MAX_DIGITS 50
#define MAX_FLIPS 1000

void generateNumber(int *arr, int maxLen);
void printArray(int *arr, int len);
int calcDecVal(int *arr, int len);
void simulateNFlips(int *flipNumber, int noOfBits, int noOfFlips, int *flips, int max);
float calcProb(int *arr, int len,int n);

int main(){

    int max,noOfBits;
    int flipNumber[MAX_DIGITS];
    int flips[MAX_FLIPS];
    float noProbs[MAX_DIGITS];
    printf(" Enter the max value of coin i.e. n : ");
    scanf("%d",&max);
    noOfBits = ceil(log(max)/log(2));
    simulateNFlips(flipNumber,noOfBits,MAX_FLIPS,flips,max);

    // printArray(flips,MAX_FLIPS);

    for(int i = 0;i<max;i++){
        printf("\n Probablity of %d: %1.3f",i+1,calcProb(flips,MAX_FLIPS,i+1));
    }
    printf("\n");
}

void simulateNFlips(int *flipNumber, int noOfBits, int noOfFlips, int *flips, int max){
    int tempStore;
    srand(time(NULL));
    for(int i=0;i<noOfFlips;i++){
```

```

        // printf("%4d) ",i+1);
        generateNumber(flipNumber,noOfBits);
        // printArray(flipNumber,noOfBits);
        // printf(" : ");
        tempStore = calcDecVal(flipNumber,noOfBits);
        if(tempStore > max || tempStore < 1){
            // printf("INVALID \n");
            i-- ;
            continue;
        }
        else{
            flips[i] = tempStore ;
            // printf("%d\n",tempStore);
        }
    }
}

void generateNumber(int *arr, int maxLen){
    for(int i = 0; i < maxLen; i++){
        arr[i] = (rand())%2;
    }
}

void printArray(int *arr, int len){
    for(int i = 0;i < len ;i++){
        printf("%d\t",arr[i]);
    }
}

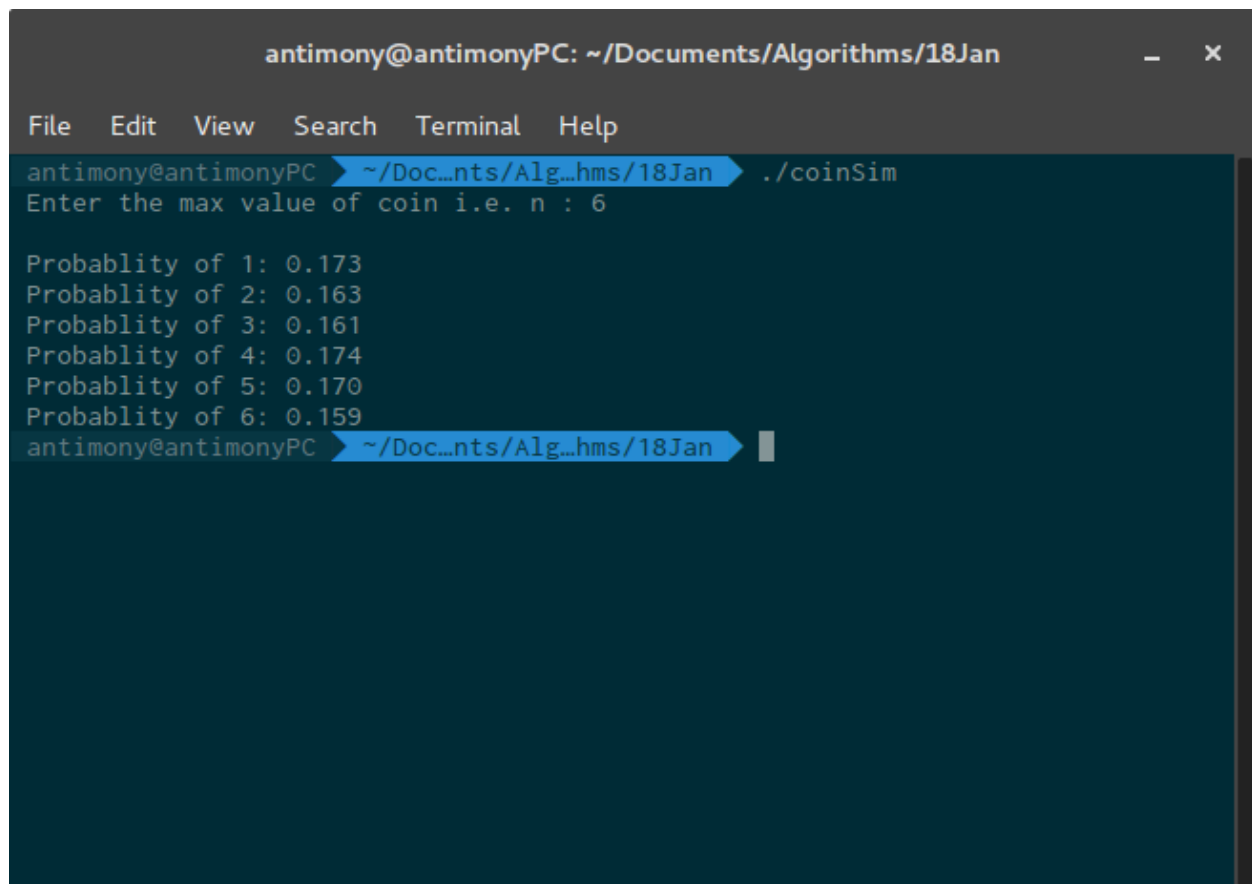
int calcDecVal(int *arr, int len){
    int i,j,sum = 0;
    for(i=len-1,j=0;i>=0,j<len;i--,j++){
        sum += arr[i]*pow(2,j);
    }
    return sum;
}

float calcProb(int *arr, int len,int n){
    int count=0;
    for(int i=0; i<len; i++){
        if(arr[i] == n){
            count++;
        }
    }
    return (double)count/(double)len;
}

```

## Result Table

Number	Probability
1	0.173
2	0.163
3	0.161
4	0.174
5	0.17
6	0.159



```
antimony@antimonyPC: ~/Documents/Algorithms/18Jan
File Edit View Search Terminal Help
antimony@antimonyPC ~/Documents/Algorithms/18Jan ./coinSim
Enter the max value of coin i.e. n : 6

Probability of 1: 0.173
Probability of 2: 0.163
Probability of 3: 0.161
Probability of 4: 0.174
Probability of 5: 0.170
Probability of 6: 0.159
antimony@antimonyPC ~/Documents/Algorithms/18Jan
```

## Analysis

Did the result meet the expectation?

Yes, the result met the expectation of around 0.16 which every number is around.

If no, can you think of an improvement?

No, I don't think any scope of improvement is there. It's almost close to the ideal value.