Q1:  
3/4  
Two points will always in a semicircle.   
Nearest: If two points lie on the same, then the p=1  
Farthest: If two points have 180 angle, then p=0.5  
Along all the points, the probability are linearly decreasing, so the expected value shoud be (1+1/2)/2=3/4.  
  
  
Q2:  
10  
Set the vertex to be A, and the opposite one to be B, set three vertex that directly connet to A to be A1, the rest to be B1.  
Use E(A) as notation for starting from A, the expected steps taken to arrive at B.   
  
E(A)=1+E(A1)  
E(A1)=1/3\*(1+E(A))+2/3(1+E(B1))  
E(B1)=1/3+2/3\*(E(A1)+1)  
  
==> E(A1)=9 E(A)=10 E(B1)=7

Q3.

Assuming there are both 26 black and red cards, obviously we would play this game, because the worst outcome would be 0 given we can always choose to stop at the end.

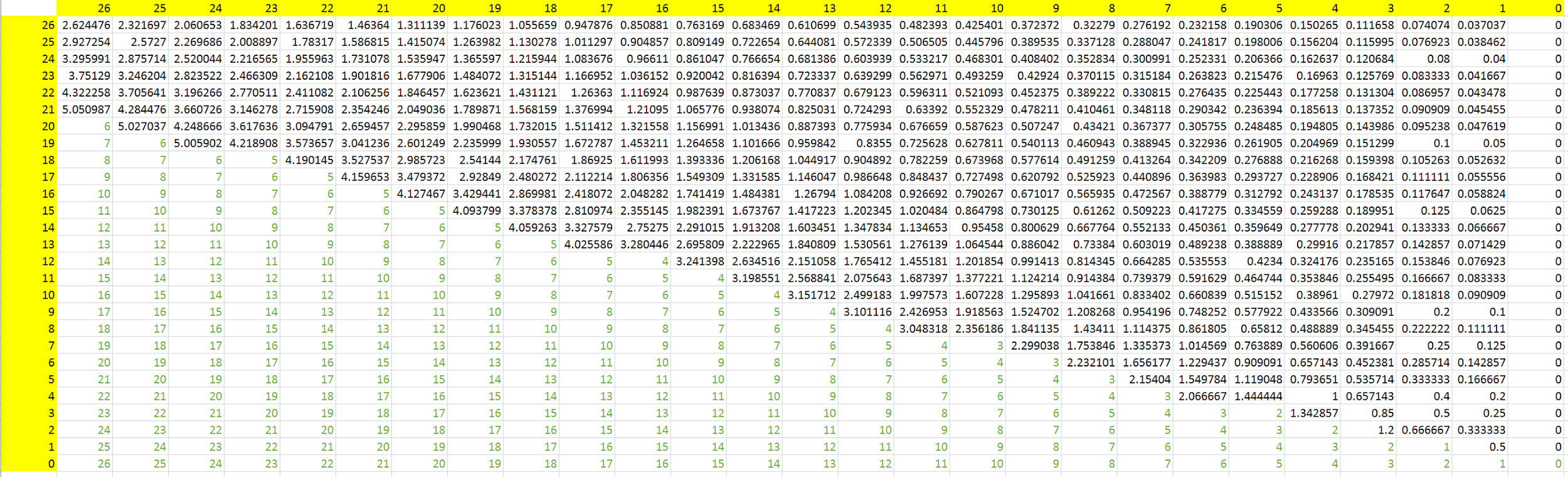
To calculate the expected payoff of this game, we can treat it like American Option pricing and price the game backwards:

We first start off at only one card left. If the card left is Black, we will have keep playing for sure and the expected payoff is 0 since all the cards is revealed. If the card left is Red, we will stop and keep $1 as the payoff.

Now let’s get to the position where there are two cards left. Using same logic we will keep playing if both left are black and get payoff 0, or stop and keep $2 as the payoff when both left are red cards. Now, if there is one each. then we need to consider two scenarios: 1. draw black, get $1 and stop. 2. draw red, get $-1 and keep playing until the end, which end up with $0 additional payoff. So the expected payoff is 0.5\*1+0.5\*0 = 0.5, you will keep play.

Using this recursive method we can calculate all the scenarios all the way back to the start. We finally get the expected payoff of the game: 2.62

Please see the excel table attached for the calculation details:



Q5.

The probability density function of the shorter lived life bulbs is f1(x)=1/100 \* e-x/100.

The probability density function of the longer lived life bulbs is f2(x)=1/200 \* e-x/200.

The survival function is 1 minus the integral of the density function (CDF). S1(x)=e-x/100, and S2(x)=e-x/200.

We can get the survival function for the entire group: (e-x/100)5 \* (e-x/200)5.

The density function for the group is the derivative of 1 minus the survival function (CDF) e-3x/40.

The expectation for the group to have first failure is the integral from 0 to infinity of x times the density function for the group = 40/3 hours.

Q6:  
Correct.  
100 year= 60\*60\*24\*365\*100 =>10^10 (seconds)  
consecutive heads=(1/2)^100=(2^10)^10=1024^10 =>10^-30  
  
P(tossing 100 consecutive heads)=P( (100 consecutive heads start from 1st second) n (100 consecutive heads start from 2nd second) n ....)<=P (100 consecutive heads) \* total time=>10^-30\*10^10=10^-20<0.01%  
  
So the statement is correct.

Q7.

Cutting a stick into N pieces with N-1 cut points is equivalent to cutting a same-length circle into N pieces with N cut points, since the first cut is arbitrary. To form a polygon, we need the length of any piece should be less than sum of other pieces. Assume the length of the stick and the circumference of the circle is 1, if one piece is more than 0.5, then they cannot form a polygon.

Define event Ei as from first cut at point i the other points are in the clockwise semicircle.

P(Ei) = , for i = 1...N.

P(cannot form a polygon) =

P(can form a polygon) =

Ref:<http://www.zhihu.com/question/25408010> and Math.pdf

Q8:  
Moving average.  
Reason:  
1. Outliers, moving average catches outliers. Outliers are signal that worth tracing.  
2. Sensitiveness. Moving average is more sensative to instant trading change, comparing to moving medium.

Q9.

We can calculate the expectation of sum of local maximas by calculating the sum of expectation of local maximas of each position.

It is not hard to see that the expectation of being maxima for the first position and last position is 0.5, since there is only one number next to it and the chance of either one been bigger is equal. We can also get that the expectation of being maxima for all the other positions is 1/3, since

there are two numbers next to it instead of one and the chance of each one been the biggest is equal.

So the answer is 0.5\*2+⅓\*(n-2)=1+(n-2)/3

Q10.

A number of power of 2 has form of 10...0 and the number minus 1 has form of 01...0. C++ Code:

bool isPower2(int x)

{

return x && !(x & (x - 1));

}

Ref: http://stackoverflow.com/questions/3638431/determine-if-an-int-is-a-power-of-2-or-not-in-a-

single-line

Q14.

C++ Code:

#include <iostream>

#include <string>

#include <vector>

using namespace std;

vector<int> indexOf(string& str, string& substr)

{

vector<int> ivec; // use vector to contain match index

int len = str.size();

int sublen = substr.size();

int limit = len - sublen + 1;

for (int i = 0; i < limit; i++)

{

int count = 0;

for (int j = 0; j < sublen; j++)

{

if (str[j+i] != substr[j])

{

i += j; // increase i by j if not match

break;

}

count++;

}

if (count == sublen)

{ ivec.push\_back(i);}

}

return ivec;

}

int main()

{

string s = "qishibuysidequantshixinhang";

string p = "shi";

vector<int> ivec = indexOf(s,p); // match at 2 and 17

for (int i = 0; i < ivec.size(); i++)

{

cout << ivec[i] << ' ';

}

return 0;

}

Q15.

C++ Code:

#include <iostream>

using namespace std;

// use Taylor expansion to approximate the e^x

double expo(double x)

{

double ret = 0;

double t = 1;

// stop approximation if the item is small enough

for (int i = 1;t > 1.0E-20;i++)

{

ret += t;

t = t \* x / i;

}

return ret;

}

int main()

{

double j = expo(20);

cout << j << endl;

return 0;

}

Q17.

Python Code:

plist = [10,11,12,13,14,12,12,11,12,14,13,11]

lp = 0 # max profit for only long

lsp = 0 # max profit for both long and short

for i in range(len(plist)-1):

if plist[i] < plist[i+1]:

lp += plist[i+1] - plist[i];

lsp += plist[i+1] - plist[i];

else:

lsp += plist[i] - plist[i+1]

print lp,lsp