Assignment 3 Solutions

COMP2804 Fall 2019

November 8, 2019

Name: Lenny Learning Combinatorics
Student ID: 100000000

1 Probabilities of Poker Hands

Let S be the set of $\binom{52}{5}$ possible hands we are dealt.

1. Let A be the event "the hand is a flush". We just need to figure out $|A|$, the number of flushes. We do that using the Product Rule with the following procedure:

   (a) Choose one of the four suits for all the cards in the flush. There are 4 ways to do this step.

   (b) Choose the ranks for the 5 cards in the flush. Since there are 13 cards whose suit matches the one we chose in the first step, there are $\binom{13}{5}$ ways to do this step.

   Therefore, the number of flushes is $|A| = 4 \cdot \binom{13}{5} = 5148$. So

   $$\Pr(A) = \frac{|A|}{|S|} = \frac{4 \cdot \binom{13}{5}}{\binom{52}{5}} = \frac{5148}{2598960} \approx 0.0019807923169267707$$

2. Let A be the event "the hand is a straight". We just need to figure out $|A|$, the number of straights. We do that using the Product Rule with the following procedure:

   (a) Select the lowest rank that takes part in the straight. This must be one of $2, 3, 4, 5, 6, 7, 8, 9, 10$, so there are 9 ways to do this step.

   (b) Select the suits of the five cards in the straight. There are five cards and 4 options for each card, so there are 4^5 ways to do this step.

   Therefore, the number of straights is $|A| = 9 \times 4^5 = 9216$. So,

   $$\Pr(A) = \frac{|A|}{|S|} = \frac{9 \cdot 4^5}{\binom{52}{5}} = \frac{9216}{2598960} \approx 0.0035460337981346383$$

3. Let A be the event "the hand is a pair". The question doesn't specify whether we should count hands that contain more than one pair, so we can accept either answer.

   (a) In case we require that the hand contains exactly one pair, we can count $|A|$ using the Product Rule as follows:

      i. Choose the rank of the pair. There are 13 options for this step.

ii. Choose the suits of the pair. There are $\binom{4}{2}$ ways to do this.

iii. Choose the remaining three cards. None of these cards should have the same rank chosen in the first step or else we will get a triple. Each of these cards should have a distinct rank or else will get a second pair. Therefore there are $\binom{12}{3}$ ways to select the ranks $\{r_1, r_2, r_3\}$ of these three cards and and 4^3 ways of choosing their suits respective suits (s_1, s_2, s_3).

Therefore, $|A| = 13 \cdot \binom{4}{2} \cdot \binom{12}{3} \cdot 4^3$. So

$$\Pr(A) = \frac{|A|}{|S|} = \frac{13 \cdot \binom{4}{2} \cdot \binom{12}{3} \cdot 4^3}{\binom{52}{5}} = \frac{1098240}{2598960} = \frac{352}{833} \approx 0.4225690276110444$$

(b) The case in which we require that the hand contain one or more pairs is (surprisingly) a little trickier. From Part (a) we know how many hands contain exactly one pair (but no triple). Now we need to add in the hands that contain two pairs (but no triple). Let B be the event "the hand contains two pairs but no triple". We can count $|B|$ using the Product Rule with the following Procedure:

i. Choose two ranks $\{r_1, r_2\}$ for the pairs. There are $\binom{13}{2}$ ways to do this.

ii. Choose the two suits for the pair with rank r_1. There are $\binom{4}{2}$ ways to do this.

iii. Choose the two suits for the pair with rank r_2. There are $\binom{4}{2}$ ways to do this.

iv. Choose the last card in the hand. The rank of this card must be different from r_1 and r_2 and can be any suit, so there are $11 \cdot 4 = 44$ ways to do this.

Therefore

$$|B| = \binom{13}{2} \cdot \binom{4}{2} \cdot \binom{4}{2} \cdot 44 = 123552 \ .$$

Now the probability we want is given by

$$\frac{|A| + |B|}{\binom{52}{5}} = \frac{1221792}{2598960} = \frac{1958}{4165} \approx 0.47010804321728694$$

These calculations require some care and it's easy to make a mistake, so we can check them with a bit of Python code:

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

from collections import defaultdict
from math import factorial
import random
import itertools

def is_flush(hand):
    suit = hand[0][1]
    for card in hand:
        if card[1] != suit:
            return False
    return True

def is_straight(hand):
    hand = sorted(hand)
    for i in range(len(hand)-1):
        if hand[i+1][0] != hand[i][0] + 1:
            return False
    return True
```

```python
def count_pairs(hand):
    d = defaultdict(int)
    pairs = 0
    for card in hand:
        d[card[0]] += 1
        if d[card[0]] == 2:
            pairs += 1
        if d[card[0]] == 3:
            return -1
    return pairs

def has_pair_a(hand):
    return count_pairs(hand) == 1

def has_pair_b(hand):
    return count_pairs(hand) >= 1

ranks = range(2, 15)
suits = [x for x in "HDCS"]

deck = list(itertools.product(ranks, suits))

c = 0
flushes = 0
straights = 0
pairs_b = 0
pairs_a = 0
for hand in itertools.combinations(deck, 5):
    flushes += is_flush(hand)
    straights += is_straight(hand)
    pairs_a += has_pair_a(hand)
    pairs_b += has_pair_b(hand)
    c += 1

print("flushes = {}".format(flushes))
print("straights = {}".format(straights))
print("pairs(a) = {}".format(pairs_a))
print("pairs(b) = {}".format(pairs_b))
```

2 Drinking Warm Beer

1. Let $X = \{M_1, \dots, M_{10}, L_1, L_2, L_3\}$ denote the set of bottles in the trunk of the car. Then the elements of S are the ordered 2-element subsets (b_1, b_2) of X. That is

$$S = \{(b_1, b_2) : b_1, b_2 \in X,\, b_1 \neq b_2\}$$

   Note that we easily determine that $|S| = 13 \cdot 12 = 156$ since we have 13 choices for b_1 and (after picking b_1) we have 12 choices for b_2.

2. This is a uniform probability space and the set S has size $13 \times 12 = 156$. Therefore $\Pr(\omega) = 1/156$ for every $\omega \in S$.

3. By a straightforward application of the Product Rule, $|A| = 10 \cdot 12 = 120$ and $|B| = 3 \times 12 = 36$, so

$$\Pr(A) = \frac{|A|}{|S|} = \frac{120}{156} = \frac{10}{13}$$

$$\Pr(B) = \frac{|B|}{|S|} = \frac{36}{156} = \frac{3}{13}$$

4. To compute $\Pr(A \mid B)$ we need to know $|A \cap B|$. Again, a straightforward application of the Product Rule tells us that $|A \cap B| = 10 \cdot 3 = 30$. So

$$\Pr(A \mid B) = \frac{\Pr(A \cap B)}{\Pr(B)} \frac{|A \cap B|/|S|}{|B|/|S|} = \frac{30}{36} = \frac{5}{6} \neq \frac{10}{13} = \Pr(A)$$

Therefore A and B are *not* independent.

3 Three Dice of a Kind

1. For this question, the sample spaces $S = \{d_1, \ldots, d_6) : d_1, \ldots, d_6 \in \{1, 2, 3, 4, 5, 6\}\}$ has size $|S| = 6^6$ and the probability space is uniform, so $\Pr(\omega) = 1/|S|$ for every $\omega \in S$.

   Let A be the event "you win this game". It's easier to consider the complementary event $\overline{A}$ and break it down into pieces:

   (a) A set $\overline{A}_1$ of outcomes in which no number appears more than once. For example $(1, 3, 5, 4, 2, 6)$. This means that $d_1, \ldots, d_6$ is a permutation of $\{1, 2, 3, 4, 5, 6\}$, so

   $$|\overline{A}_1| = 6! \ .$$

   (b) A set $\overline{A}_{2,1}$ of outcomes in which exactly one number appears twice. For example $(2, 1, \underline{3}, 4, \underline{3}, 6)$. There are $\binom{6}{2}$ choices for the locations of this number, 6 choices for the value of this number, and then $5 \cdot 4 \cdot 3 \cdot 2$ choices for the values of the other 4 dice. Therefore

   $$|\overline{A}_{2,1}| = \binom{6}{2} \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \ .$$

   In the example $(2, 1, \underline{3}, 4, \underline{3}, 6)$ we first choose the locations 3 and 5 so we have $(\cdot, \cdot, x, \cdot, x, \cdot)$. Then we choose the value 3 so we have $(\cdot, \cdot, 3, \cdot, 3, \cdot)$. Then we choose the values $2, 1, 4, 6$ so we have $(2, 1, 3, 4, 3, 6)$.

   (c) A set $\overline{A}_{2,2}$ of outcomes in which exactly two numbers x_1, x_2 appear twice. For example $(\underline{1}, \underline{3}, 5, \underline{1}, 4, \underline{3})$. There are $\binom{6}{2}$ choices for the the values of these two numbers. Call these values x_1 and x_2 where $x_1 > x_2$. There are $\binom{6}{2}$ choices for the locations of x_1 after which there are $\binom{4}{2}$ choices for the locations of x_2. Finally, there 4×3 choices for the values of the numbers that go in the remaining two locations. Therefore,

   $$|\overline{A}_{2,2}| = \binom{6}{2} \cdot \binom{6}{2} \cdot \binom{4}{2} \cdot 4 \cdot 3 \ .$$

   In the example $(\underline{1}, \underline{3}, 5, \underline{1}, 4, \underline{3})$ we first choose the values $x_1 = 3$ and $x_2 = 1$. Then we choose the locations 2 and 6 for x_1 giving $(\cdot, 3, \cdot, \cdot, \cdot, 3)$. Then we choose the locations 1 and 4 for x_2 giving $(1, 3, \cdot, 1, \cdot, 3)$. Finally we choose the values $5, 4$ giving $(1, 3, 5, 1, 4, 3)$.

   (d) A set $\overline{A}_{2,3}$ of outcomes in which three numbers x_1, x_2, x_3 each appear twice. For example $(1, 4, 4, 2, 1, 2)$. There are $\binom{6}{3}$ choices for $x_1 > x_2 > x_3$. Then there are $\binom{6}{2}$ choices for the locations of x_1. Then $\binom{4}{2}$ choices for the locations of x_2. This leaves only $\binom{2}{2} = 1$ choices for the locations of x_3. Therefore,

   $$|\overline{A}_{2,3}| = \binom{6}{3} \cdot \binom{6}{2} \cdot \binom{4}{2} \ .$$

In the example $(1,4,4,2,1,2)$ we choose $x_1 = 4$, $x_2 = 2$, and $x_1 = 1$. Then we choose $2,3$ as the locations for x_1 giving $(\cdot,4,4,\cdot,\cdot,\cdot)$. Then we choose $4,6$ as the locations for x_2 giving $(\cdot,4,4,2,\cdot,2)$. Then we have no choice but to place x_3 at positions $1,5$ giving $(1,4,4,2,1,2)$.

Since $\overline{A} = \overline{A}_1 \cup \overline{A}_{2,1} \cup \overline{A}_{2,2} \cup \overline{A}_{2,3}$ and $\overline{A}_1$, $\overline{A}_{2,1}$, $\overline{A}_{2,2}$ and $\overline{A}_{2,3}$ are pairwise disjoint, we get

$$\begin{aligned}|\overline{A}| &= |\overline{A}_1| + |\overline{A}_{2,1}| + |\overline{A}_{2,2}| + |\overline{A}_{2,3}| \\ &= 6! + \binom{6}{2}\cdot 6\cdot 5\cdot 4\cdot 3\cdot 2 + \binom{6}{2}\cdot\binom{6}{2}\cdot\binom{4}{2}\cdot 4\cdot 3 + \binom{6}{3}\cdot\binom{6}{2}\cdot\binom{4}{2} \\ &= 29520\end{aligned}$$

Therefore,

$$\Pr(A) = 1 - \Pr(\overline{A}) = \frac{6^6 - 29520}{6^6} = \frac{17136}{46656} = \frac{199}{324} \approx 0.36728395061728397$$

This was an involved computation with lots of chances for calculation errors or double-counting, so here's some code to check it by exhaustive enumeration:

```
#!/usr/bin/python3

from collections import defaultdict
from math import factorial
from fractions import Fraction
import itertools

def binom(n, k):
    return factorial(n)//(factorial(k)*factorial(n-k))

def win(a):
    d = defaultdict(int)
    for x in a:
        d[x] += 1
        if d[x] == 3:
            return True
    return False

def count_winners():
    w = 0
    die = [1, 2, 3, 4, 5, 6]
    for p in itertools.product(die, die, die, die, die, die):
        w += win(p)
    return w

if __name__ == "__main__":
    s = 6**6
    w = count_winners()
    print("Python: Pr(A) = {}/{} = {} ~ {}".format(w, s, Fraction(w,s), w/s))

    nota = \
        factorial(6) \
        + binom(6,2)*6*5*4*3*2 \
        + binom(6,2)*binom(6,2)*binom(4,2)*4*3 \
        + binom(6,3)*binom(6,2)*binom(4,2)
    a = s - nota
```

```
pa = Fraction(a, s)
print("Brain: Pr(A) = {} ~ {} ".format(pa, float(pa)))
```

4 Dungeons and Pepys

1. Let A be the event "we win Game A". For each $i \in \{1, \ldots, 12\}$, let A_i be the event "the ith roll is a 12". Then

$$\begin{aligned}\Pr(\overline{A} &= \Pr(\overline{A}_1 \cap \overline{A}_2 \cap \cdots \overline{A}_{12}) \\ &= \Pr(\overline{A}_1) \cdot \Pr(\overline{A}_2) \cdot \cdots \cdot \Pr(\overline{A}_{12}) \\ &= \left(\frac{11}{12}\right)^{12} \\ &= \frac{3138428376721}{8916100448256}\end{aligned}$$

So

$$\Pr(A) = 1 - \Pr(\overline{A}) = \frac{5777672071535}{8916100448256} \approx 0.6480043719858629 \ .$$

2. Let B be the event "we win Game B" so $\overline{B}$ is the event "we lose Game B". For each $i \in \{0, 1, 2\}$, let X_i be the event "we roll exactly i twelves". Then

$$\begin{aligned}\Pr(\overline{B}) &= \Pr(X_0 \cup X_1 \cup X_2) \\ &= \Pr(X_0) + \Pr(X_1) + \Pr(X_2) \\ &= \left(\frac{11}{12}\right)^{36} + 36 \cdot \frac{1}{12} \cdot \left(\frac{11}{12}\right)^{35} + \binom{36}{2} \cdot \left(\frac{1}{12}\right)^{2} \cdot \left(\frac{11}{12}\right)^{34} \\ &= \frac{2930317733157244753150193042860323058 27}{70880187498509184538134430700956916121 6}\end{aligned}$$

So

$$\Pr(B) = 1 - \Pr(\overline{B}) = \frac{415770101669367370066325002723536855389}{708801874985091845381344307009569161216} \approx 0.5865815488680983 \ .$$

(That number 0.5865815488680983 looks familiar. Maybe we'll see it again...)

5 Random Pigeonholing

1. This is just the Birthday Paradox computation. The outcome set S consists of all functions from a set of size 100 onto a set of size 500 and therefore $|S| = 500^{100}$. Let $\overline{A}$ be the event "every hole contains at most one pigeon". Then $\overline{A}$ consists of all one-to-one functions from a set of size 100 onto a set of size 500 and therefore $|\overline{A}| = \frac{500!}{400!}$. So

$$\Pr(\overline{A}) = \frac{|\overline{A}|}{|S|} = \frac{500!}{400! \cdot 500^{100}} \ .$$

So

$$\Pr(A) = 1 - \Pr(\overline{A}) \approx 0.9999758457295991 \ .$$

2. Again, it's easier (but not much) to work with the complimentary event. Let B be the event "at least one hole contains at least 3 pigeons" so that $\overline{B}$ is the event "every hole contains at most 2 pigeons". For each $k \in \{0, \ldots, 50\}$, let $\overline{B}_k$ denote the event "every hole contains at most 2 pigeons and exactly

k holes contain exactly two pigeons". Notice that $\overline{A} = \overline{B}_0$ (where $\overline{A}$ is defined in the answer to the previous question). $B_0, \ldots, B_{50}$ are pairwise-disjoint and $\overline{B} = \overline{B}_0 \cup \overline{B}_1 \cup \cdots \cup \overline{B}_{50}$, so

$$|\overline{B}| = |\overline{B}_0 \cup \overline{B}_1 \cup \cdots \cup \overline{B}_{50}| = |\overline{B}_0| + |\overline{B}_1| + \cdots + |\overline{B}_{50}|$$

To figure out $|\overline{B}_k|$ we'll use the Product Rule with a 2-step procedure.

(a) First, let's choose the k-pairs of pigeons that will fly into holes together. The number of ways of choosing k pigeons and pairing them off is exactly the number of self-inverting functions on a set of size 100 that have exactly $100 - 2k$ fixed points. From Assignment 2, we know there are

$$\binom{100}{2k}\left(\frac{1}{2^k}\right)\binom{2k}{k}k!$$

ways to do this.

(b) Now that we've chosen the k pairs of pigeons we need to pick a distinct hole for each of the k pairs and for each of the $100 - 2k$ lonely pigeons. So we're looking for a one-to-one function from a set of size $100 - k$ onto a set of size 500. There are $500!/(400 + k)!$ such functions.

So by the Product Rule,

$$|\overline{B}_k| = \binom{100}{2k}\left(\frac{1}{2^k}\right)\binom{2k}{k}k! \cdot \frac{500!}{(400+k)!}$$

So

$$\begin{aligned} |\overline{B}| &= \sum_{k=0}^{50} |\overline{B}_k| \\ &= \sum_{k=0}^{50} \binom{100}{2k}\left(\frac{1}{2^k}\right)\binom{2k}{k}k! \cdot \frac{500!}{(400+k)!} \\ &= 44481511508\ldots41760000000000000000000000000 \end{aligned}$$

(there are many digits missing under the $\ldots$). We finish up with

$$\begin{aligned} \Pr(B) &= 1 - \Pr(\overline{B}) \\ &= 1 - \frac{|\overline{B}|}{500^{100}} \\ &= 3440457901373742106810630796692352080135529819446700830244386\,56 \\ &\quad 1213886717857931461159732276036555508814363371919495444982141\,32 \\ &\quad 2132426982325128266736576998773588934146828095306706070495971\,69 \\ &\quad 3952255046360928879602096617988671000395858029054420582400000\,00 \\ &\quad 000000000000000000/500^{100} \approx 0.4361298523736379 \end{aligned}$$

6 Rolling Snake Eyes

1. This is a uniform probability space of the set $S = \{(d_1, d_2) : d_1, d_2 \in \{1, 2, 3, 4, 5, 6\}\}$. By the Product Rule, $|S| = 1/36$. The event $A = \{(1, 1)\}$ has size 1, so $\Pr(A) = 1/|S| = 1/36$.

2. Let B be the event "at least one of our dice is a 1". Then

$$B = \{(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1)\}$$

and $|B| = 11$. So

$$\Pr(A \mid B) = \frac{\Pr(A \cap B)}{Pr(B)} = \frac{\Pr(\{(1, 1)\})}{\Pr(B)} = \frac{1/36}{11/36} = \frac{1}{11}.$$

3. Let C be the event "at least one of the dice came up 6". Just like B, C has size 11. More importantly, $C \cap B = \{(1,6),(6,1)\}$. So

$$\Pr(C \mid B) = \frac{\Pr(B \cap C)}{\Pr(B)} = \frac{2/36}{11/36} = \frac{2}{11} \ .$$

4. Here $S = \{(d_1, d_2, d_3, d_4) : d_1, d_2, d_3, d_4 \in \{1,2,3,4,5,6\}\}$ has size $|S| = 6^4$.

   By an easy application of the Product Rule, $|X| = |Y| = 36$. Obviously $X \cap Y = \{(1,1,1,1)\}$ has size $|X \cap Y| = 1$. Let $Z = X \cup Y$. By principle of inclusion-exclusion:

$$|Z| = |X \cup Y| = |X| + |Y| - |X \cap Y| = 36 + 36 - 1 = 71 \ .$$

   Let A be the event "$d_1 = d_2 = 1$" so that the question is asking us to compute $\Pr(A \mid Z)$.

   So
$$A \cap Z = \{(1,1,1,d) : d \in \{1,2,3,4,5,6\}\} \cup \{(1,1,d,1) : d \in \{1,2,3,4,5,6\}\} \ .$$

   By the Principle of Inclusion-Exclusion

$$|A \cap Z| = 6 + 6 - 1 = 11 \ .$$

   Therefore,
$$\Pr(A \mid Z) = \frac{\Pr(A \cap Z)}{\Pr(Z)} = \frac{11/6^4}{71/6^4} = \frac{11}{71} \ .$$

7 Randomized Leader Election

1. For the case $n = 2$, we have we have a uniform probability space over the sample set $S = \{(1,1),(1,2),(2,1),(2,2)\}$. If we let A_2 denote the event "the algorithm succeeds" then $A = \{(1,2),(2,1)\}$ so

$$\Pr(A) = \frac{|A|}{|S|} = \frac{2}{4} = \frac{1}{2} \ .$$

2. For the case $n = 3$, we have a uniform probability space over $S = \{1,2,3\}^3$, which has size $3^3 = 27$. Let A denote the event "the algorithm succeeds". For each $i \in \{1,2,3\}$, let A_i denote the event "the algorithm succeeds and the maximum is i". Then $A_1 = \emptyset$ since the algorithm does not succeed if all three players choose 1. Next,
$$A_2 = \{(2,1,1),(1,2,1),(1,1,2)\} \ ,$$
   so $|A_2| = 3$. Finally,

$$A_3 = \{(3,a,b) : a,b \in \{1,2\}\} \cup \{(a,3,b) : a,b \in \{1,2\}\} \cup \{(a,b,3) : a,b \in \{1,2\}\} \ ,$$

   so A_3 is the union of three disjoint sets, each of size 4. Therefore $|A_3| = 12$. We finish up with

$$\Pr(A) = \Pr(A_1 \cup A_2 \cup A_3) = \Pr(A_1) + \Pr(A_2) + \Pr(A_3) = \frac{0 + 3 + 12}{27} = \frac{15}{27} \ .$$

3. For the general case, we can proceed as we did for the Case $n = 3$. Let A be the event "the algorithm succeeds". For each $i \in \{1, \ldots, n\}$, let A_i be the event "the algorithm succeeds and the maximum is i". Then
$$|A_i| = n \times (i-1)^{n-1} \ ,$$

since we need to choose a location for i to occur and then the remaining $n-1$ locations need to be filled with choices from $1, \ldots, i-1$. So

$$\begin{aligned}
\Pr(A) &= \Pr(A_1 \cup A_2 \cup \cdots \cup A_n) \\
&= \sum_{i=1}^{n} \Pr(A_i) \\
&= \sum_{i=1}^{n} \frac{|A_i|}{|S|} \\
&= \sum_{i=1}^{n} \frac{n \times (i-1)^{n-1}}{n^n} \\
&= \sum_{i=1}^{n} \left(\frac{(i-1)}{n})\right)^{n-1} \\
&= \sum_{i=0}^{n-1} \left(\frac{i}{n}\right)^{n-1} \\
&= \sum_{i=1}^{n-1} \left(\frac{i}{n}\right)^{n-1}
\end{aligned}$$

We can't say anything more precise than this about $\Pr(A)$, and I don't expect anyone to get beyond this. In particular, there's no closed form for the sum in the numerator, but we can approximate it very closely. Consider

$$\begin{aligned}
\Pr(A) &= \sum_{i=1}^{n-1} (i/n)^{n-1} \\
&= \left(\frac{n-1}{n}\right)^{n-1} + \left(\frac{n-2}{n}\right)^{n-1} + \left(\frac{n-3}{n}\right)^{n-1} + \cdots + \left(\frac{n-(n-1)}{n}\right)^{n-1} \\
&= (1-1/n)^{n-1} + (1-2/n)^{n-1} + (1-3/n)^{n-1} + \cdots + (1-(n-1)/n)^{n-1} \\
&= \sum_{i=1}^{n-1} (1-i/n)^{n-1} \\
&= \sum_{i=1}^{n-1} \left(\left((1-i/n)^{\frac{n}{i}}\right)^{\frac{n-1}{n}}\right)^{i} \\
&\leq \sum_{i=1}^{n-1} \left((1/e)^{\frac{n-1}{n}}\right)^{i} \\
&\leq \sum_{i=1}^{\infty} \left((1/e)^{\frac{n-1}{n}}\right)^{i} \\
&= \frac{1}{e^{\frac{n-1}{n}} - 1}
\end{aligned}$$

Notice that, as $n \to \infty$, this converges to $\Pr(A) \leq 1/(e-1) \approx 0.581976$. If we want a matching lower bound then, for large n, we can continue from partway down:

$$\begin{aligned}\Pr(A) &= \sum_{i=1}^{n-1} \left(\left((1 - i/n)^{\frac{n}{i}} \right)^{\frac{n-1}{n}} \right)^i \\ &\geq \sum_{i=1}^{\sqrt{n}} \left(\left((1 - i/n)^{\frac{n}{i}} \right)^{\frac{n-1}{n}} \right)^i \\ &\geq \sum_{i=1}^{\sqrt{n}} \left(\left((1 - 1/\sqrt{n})^{\sqrt{n}} \right)^{\frac{n-1}{n}} \right)^i\end{aligned}$$

since $f(k) = (1 - 1/k)^k$ is an increasing function. Now, $\lim_{n\to\infty}((1 - 1/\sqrt{n})^{\sqrt{n}})^{\frac{n-1}{n}} = 1/e$ and

$$\sum_{i=1}^{\sqrt{n}} (1/e)^i = \frac{1 - 1/e^{\sqrt{n}}}{e - 1} \to \frac{1}{e-1} \quad \text{as } n \to \infty.$$

Doing this carefully enough, we can conclude that

$$\lim_{n\to\infty} \Pr(A) = \frac{1}{e-1} \approx 0.581976$$

So for large values of n the algorithm succeeds in electing a leader about 58% of the time.

We can also check the sanity of this result by simulation:

```python
#!/usr/bin/python3
import random
import sys


if __name__ == "__main__":
    n = 100
    if len(sys.argv) > 1:
        n = int(sys.argv[1])

    print("Estimating Pr(A) for n={}".format(n))
    i = 0
    c = 0
    while 1 < 2:
        a = [random.randrange(n) for _ in range(n)]
        m = max(a)
        if len([x for x in a if x == m]) == 1:
            c += 1
        i += 1
        print("{} {} {}    \r".format(c, i, c/i), end='')
```