

Randomness and Computation  
Coursework 2  
UNN: S1796157

## Question 1

a)

To prove the equality, we need to define some other basic statements which will be useful during the proof.

1. We defined  $|AO(G)|$  as the number of DAGs which can be constructed from  $G$ . We also defined  $A_{n,m}$  as the total number of DAGs which have  $n$  vertices and  $m$  edges. We now redefine  $A_{n,m}$  with respect to  $AO(G)$ :

$$A_{n,m} = \sum_{G=(V,E), |V|=n, |E|=m} |AO(G)| \quad (1)$$

$A_{n,m}$  can be seen as the sum of all  $|AO(G)|$  over all possible undirected graphs  $G$  with exactly  $m$  edges and  $n$  vertices;

2. We also defined  $\mathbb{E}[|AO(G)|] = \sum_{G=(V,E), |V|=n} Pr[G] \cdot |AO(G)|$ , which is the expected number of AOs given a graph generated from  $G_{n,p}$ . We now redefine the expectation to take into account the number of edges:

$$\mathbb{E}[|AO(G)|] = \sum_{m=0}^{\binom{n}{2}} \sum_{\substack{G=(V,E) \\ |V|=n, |E|=m}} Pr[G] \cdot |AO(G)| \quad (2)$$

This new way to express the expectation is equivalent to the original definition. For a given number of edges  $m$ , we now sum the product  $P[G] \cdot |AO(G)|$  over all possible graphs  $G$  which have  $|E| = m$ . We can proceed further since we know that all graphs with  $m$  edges have the same probability in the  $G_{n,p}$  model. Therefore:

$$\mathbb{E}[|AO(G)|] = \sum_{m=0}^{\binom{n}{2}} Pr[G] \sum_{\substack{G=(V,E) \\ |V|=n, |E|=m}} |AO(G)| \quad (3)$$

*Proof.*

$$\mathcal{A}_n \left( \frac{p}{1-p} \right) = (1-p)^{-\binom{n}{2}} \cdot \mathbb{E}[|AO(G)|] \quad (4)$$

$$\sum_{m=0}^{\binom{n}{2}} \mathcal{A}_{n,m} \left( \frac{p}{1-p} \right)^m = (1-p)^{-\binom{n}{2}} \cdot \sum_{m=0}^{\binom{n}{2}} Pr[G] \sum_{\substack{G=(V,E) \\ |V|=n, |E|=m}} |AO(G)| \quad (5)$$

$$\sum_{m=0}^{\binom{n}{2}} \mathcal{A}_{n,m} \cdot p^m \cdot (1-p)^{-m} = (1-p)^{-\binom{n}{2}} \sum_{m=0}^{\binom{n}{2}} Pr[G] \sum_{\substack{G=(V,E) \\ |V|=n, |E|=m}} |AO(G)| \quad (6)$$

$$\sum_{m=0}^{\binom{n}{2}} \mathcal{A}_{n,m} \cdot p^m \cdot (1-p)^{-m} = (1-p)^{-\binom{n}{2}} \sum_{m=0}^{\binom{n}{2}} p^m (1-p)^{\binom{n}{2}} \cdot (1-p)^{-m} \sum_{\substack{G=(V,E) \\ |V|=n, |E|=m}} |AO(G)| \quad (7)$$

$$\sum_{m=0}^{\binom{n}{2}} \mathcal{A}_{n,m} \cdot p^m \cdot (1-p)^{-m} = \cancel{(1-p)^{\binom{n}{2}}} \cancel{\binom{n}{2}} \sum_{m=0}^{\binom{n}{2}} p^m \cdot (1-p)^{-m} \cdot \sum_{\substack{G=(V,E) \\ |V|=n, |E|=m}} |AO(G)| \quad (8)$$

$$\sum_{m=0}^{\binom{n}{2}} \mathcal{A}_{n,m} \cdot p^m \cdot (1-p)^{-m} = \sum_{m=0}^{\binom{n}{2}} p^m \cdot (1-p)^{-m} \cdot \sum_{\substack{G=(V,E) \\ |V|=n, |E|=m}} |AO(G)| \quad (9)$$

From equation (1) we know that  $\mathcal{A}_{n,m} = \sum_{G=(V,E), |V|=n, |E|=m} |AO(G)|$  and therefore equation (9) becomes:

$$\sum_{m=0}^{\binom{n}{2}} \mathcal{A}_{n,m} \cdot p^m \cdot (1-p)^{-m} = \sum_{m=0}^{\binom{n}{2}} p^m \cdot (1-p)^{-m} \cdot \mathcal{A}_{n,m} \quad (10)$$

Hence, the equality is proven.  $\square$

b)

To evaluate  $\mathbb{E}[|AO(G)|]$  we just need to compute the  $A(\frac{p}{1-p}) \cdot (1-p)^{\binom{n}{2}}$  value (we can trivially derive it from the equality proven in the previous question). We will also employ the recurrence given below, which can be efficiently computed by using dynamic programming:

$$\mathcal{A}_n \left( \frac{p}{1-p} \right) = \sum_{i=1}^n (-1)^{i+1} \binom{n}{i} \left( 1 + \frac{p}{1-p} \right)^{i(n-i)} \mathcal{A}_{n-i} \left( \frac{p}{1-p} \right) \quad (11)$$

$$= \sum_{i=1}^n (-1)^{i+1} \binom{n}{i} \left( \frac{1}{1-p} \right)^{i(n-i)} \mathcal{A}_{n-i} \left( \frac{p}{1-p} \right) \quad (12)$$

We now show the pseudocode:

---

**Algorithm 1** Expectation algorithm for graphs  $G$  generated from the Erdős-Rényi model  $G_{n,p}$

---

```

1: function EXPAO(n,p)
2:    $A \leftarrow \text{array}[0 \dots n]$ ; ▷ This array will store the  $\mathcal{A}_n$  values.
3:    $pows \leftarrow \text{array}[0 \dots \lceil \frac{n^2}{4} \rceil]$ ; ▷ This array will store the  $\left( \frac{1}{1-p} \right)^{i(n-i)}$  values.
4:    $pows[0] \leftarrow 1$ ;
5:   for  $i \leftarrow 1$  to  $\lceil \frac{n^2}{4} \rceil$  do ▷  $\lceil \frac{n^2}{4} \rceil$  is the highest exponent we will find.
6:      $pows[i] \leftarrow pows[i-1] \cdot \frac{1}{1-p}$ ;
7:   end for
8:    $A[0] \leftarrow A[1] \leftarrow 1$ ; ▷ Base cases for dynamic programming.
9:   for  $i \leftarrow 2$  to  $n$  do
10:     $A[i] \leftarrow 0$ ;
11:     $b \leftarrow 1$ ;
12:    for  $j \leftarrow 1$  to  $i$  do
13:       $b \leftarrow \frac{b \cdot (i-j+1)}{i}$ ; ▷ Here we compute the binomial coefficient.
14:       $sign \leftarrow \text{if } (j+1)\%2 == 0 \text{ then } 1 \text{ else } -1$ ; ▷ We check the sign of 1.
15:       $A[i] \leftarrow A[i] + sign \cdot b \cdot pows[j \cdot (i-j)] \cdot A[i-j]$ ;
16:    end for
17:  end for
18:  return  $A[n] \cdot (1-p)^{\binom{n}{2}}$ 
19: end function

```

---

The resulting algorithm has a complexity of  $\mathcal{O}(n^2)$ . Here it is assumed that the exponentiation of a number, namely  $b^k$ , has a complexity of  $\mathcal{O}(k)$  (we assume it is obtained by multiplying  $k$  times  $b$ ). We can easily derive the  $\mathcal{O}(n^2)$  upper bound noticing that:

1. On line 5, to compute all the powers of  $\frac{1}{1-p}$  we need to perform  $\lceil \frac{n^2}{4} \rceil$  operations, which leads to a complexity of  $\mathcal{O}(n^2)$ ;
2. On line 9, the double for loop runs for  $\sum_{i=1}^{n-2} (n-i)i$  times, which clearly has a complexity of  $\mathcal{O}(n^2)$ ;
3. On line 18, computing the quantity  $(1-p)^{\binom{n}{2}}$  has a complexity of  $\mathcal{O}(\binom{n}{2}) = \mathcal{O}(n^2)$ .

Therefore, the total complexity of the algorithm is  $\mathcal{O}(3n^2)$ , hence we can assume that is  $\mathcal{O}(n^2)$ .

## Question 2

The derandomization of the modified Max-Cut algorithm works as follows. We imagine placing vertices  $v_i, \dots, v_n$  deterministically, one at a time, into the partition  $L$ , which has to have maximum size of  $\lfloor \frac{n}{2} \rfloor$ . Suppose we have already placed the first  $k$  vertices in  $L$  (with  $k < \lfloor \frac{n}{2} \rfloor$ ), and consider the expected value of the cut if we choose randomly other  $\lfloor \frac{n}{2} \rfloor - k$  vertices and we place them into  $L$ . This quantity is specified as follow  $\mathbb{E}[C(L, V \setminus L) | \{v_1, \dots, v_k\} \in L]^1$  and it represents the expected size of the cut, given that we know that  $L$  contains the vertices  $\{v_1, \dots, v_k\}$ . We now show how to select the next vertices which will be placed into  $L$  such that:

$$\mathbb{E}[C(L, V \setminus L) | \{v_1, \dots, v_k\} \in L] \leq \mathbb{E}[C(L, V \setminus R) | \{v_1, \dots, v_{k+1}\} \in L] \quad (13)$$

It follows that:

$$\mathbb{E}[C(L, V \setminus L)] \leq \mathbb{E}[C(L, V \setminus L) | \{v_1, \dots, v_{\lfloor \frac{n}{2} \rfloor}\} \in L] \quad (14)$$

The right side of the inequality shows the size of the cut when we have completely filled the  $L$  partition (if the  $L$  partition is full, then we consider all the remaining vertices to be on the “other side” of the cut and our algorithms has ended). Therefore, we can ultimately say that:

$$\mathbb{E}[C(L, V \setminus L) | \{v_1, \dots, v_{\lfloor \frac{n}{2} \rfloor}\} \in L] \geq m \frac{n}{2n-1} \quad (15)$$

*Proof.* We can prove the previous statements by induction. The base case in the induction is:

$$\mathbb{E}[C(L, V \setminus L) | \{\emptyset\} \in L] \geq m \frac{n}{2n-1} \quad (16)$$

Then, we prove the induction step, which is:

$$\mathbb{E}[C(L, V \setminus L) | \{v_1, \dots, v_k\} \in L] \leq \mathbb{E}[C(L, V \setminus L) | \{v_1, \dots, v_{k+1}\} \in L] \quad (17)$$

We consider now what of the remaining  $n - k$  vertices has to be included into the  $L$  partition. We want them to have an uniform probability of being selected. Therefore, the expectation is modeled as:

$$\mathbb{E}[C(L, V \setminus L) | \{v_1, \dots, v_k\} \in L] = \sum_{v \in V \setminus L} \frac{1}{n-k} \mathbb{E}[C(L, V \setminus L) | \{v_1, \dots, v_k, v\} \in L] \quad (18)$$

It follows that:

$$\max_{v \in V \setminus L} (\mathbb{E}[C(L, V \setminus L) | \{v_1, \dots, v_k, v\} \in L]) \geq \mathbb{E}[C(L, V \setminus L) | \{v_1, \dots, v_k\} \in L] \quad (19)$$

Therefore, we just have to compute which one of the  $v \in V \setminus L$  gives the largest expectation and then we can add it to the  $L$  partition. Once we do this, we will define a placement which satisfies:

$$\mathbb{E}[C(L, V \setminus L) | \{v_1, \dots, v_k\} \in L] \leq \mathbb{E}[C(L, V \setminus L) | \{v_1, \dots, v_{k+1}\} \in L] \quad (20)$$

□

---

<sup>1</sup> $C(L, R)$  is a function which returns the cut size given two vertices partitions,  $L$  and  $R$ , of a graph  $G$ , such that  $L \cap R = \{\emptyset\}$ .

We now describe how to compute the expectation  $\mathbb{E}$  efficiently, given that we know a subset of  $L$ . Since the expectation computes the expected size of the cut, we can represent it as the expected number of edges which will be part of the cut.  $\mathbb{E}$  can be written as:

$$\mathbb{E}[C(L, V \setminus L) | \{v_1, \dots, v_k\} \in L] = |e_L| \cdot P_{e_L} + (|E| - |e_L|) \cdot P_{|E| - e_L} \quad (21)$$

We now explain the terms inside the expectation formula:

1. By  $e_L$  we indicates the number of edges which have a vertex  $v \in L$  and therefore can be part of the cut. This number can be computed efficiently everytime we add a new vertex into  $L$ .
2. Each edge which has one of its vertices already inside  $L$  has the same probability to being selected for the final cut. Namely, the other vertex has never to be selected for the cut, and this event has probability:

$$P_{e_L} = \prod_{i=0}^{\lfloor \frac{n}{2} \rfloor - k} \frac{n - k - i - 1}{n - k - i} \quad (22)$$

3. By  $|E| - e_L$  we indicates the remaining number of edges which can be still selected to be part of the cut. Each of them has the same probability of being selected. Namely, the probability is computed as follow:

$$P_{|E| - e_L} = 2 \cdot \sum_{j=0}^{\lfloor \frac{n}{2} \rfloor - k} \frac{1}{n - k - j} \prod_{i=0}^{\lfloor \frac{n}{2} \rfloor - k} \frac{n - k - i - 1}{n - k - i} \quad (23)$$

Given an edge  $e$  composed by vertices  $u$  and  $v$ , the formula above compute the probability that just one of the two vertices is selected to be placed into  $L$  and the other is never selected (the values is doubled since its symmetric with respect to the vertex which is choosen).

The expectation can be computed in  $\mathcal{O}(n)$  time, since computing  $P_e$  and  $P_{|E| - e_L}$  takes linear time (we can notice how the product of sequence of term in  $P_{|E| - e_L}$  is equal to  $P_e$  and therefore it can be computed only once).

We now provide the pseudocode for the derandomized Max-Cut algorithm:

---

**Algorithm 2** Derandomized Max-Cut algorithm

---

```

1: function MAX-CUT( $G = (V, E)$ )
2:    $L \leftarrow \{\}$ ;
3:   for  $i = 0$  to  $\lceil \frac{n}{2} \rceil$  do
4:      $maxExp \leftarrow -1$ ;
5:      $newState \leftarrow \{\}$ ;
6:     for  $v \in V \setminus L$  do
7:        $nextExp = \mathbb{E}(C(L, R) | L \cup v)$ ;            $\triangleright$  Expectation we would ge by adding  $v$  to  $L$ .
8:       if  $nextExp > maxExp$  then
9:          $nextExp \leftarrow maxExp$ ;
10:         $newState \leftarrow \{v\}$ ;
11:      end if
12:    end for
13:     $L \leftarrow L \cup v$ ;
14:  end for
15:  return  $L$ 
16: end function

```

---

We now analyze its complexity:

1. On line 3, the loop has a complexity of  $\mathcal{O}(\lceil \frac{n}{2} \rceil) = \mathcal{O}(n)$ ;
2. On line 6, we loop over all vertices and therefore it has complexity of  $\mathcal{O}(n)$ ;

3. On line 7, computing the expectations requires linear time with respects of the number of vertices and therefore it has complexity  $O(n)$ ;

In conclusion, we expect this algorithm to have a final complexity of  $\mathcal{O}(n^3)$ .

### Question 3

a)

We procede by first rewriting the original statement into a form which will be more useful to us later on. With  $S$  we indicate the number of samples (namely, how many times we sampled from  $Y$ ). With  $Z$  we denote the new random variable obtained by sampling  $S$  time from  $Y$  and with  $X$  we indicate the number of time we got  $Y = 1$  (which is the sum off all samples we got, namely  $Z = \sum_{i=0}^S X_i = X$ ).

$$P[|\hat{p} - p| \leq \frac{1}{N}] \geq 1 - \frac{1}{N} \quad (24)$$

$$P[|\frac{X}{S} - p| \leq \frac{1}{N}] \geq 1 - \frac{1}{N} \quad (25)$$

$$P[|X - Sp| \leq \frac{S}{N}] \geq 1 - \frac{1}{N} \quad (26)$$

$$1 - P[|X - Sp| > \frac{S}{N}] \geq 1 - \frac{1}{N} \quad (27)$$

$$P[|X - Sp| > \frac{S}{N}] \leq \frac{1}{N} \quad (28)$$

We can easily notice that the coefficient  $Sp$  is equivalent to the expected value of  $Z$ . We can now proceed by using the Chernoff's bound to find the minimal number of samples needed to satisfies inequality (24). We will use this version of the Chernoff's bound:

$$P[|X - \mu| \geq \sigma\delta] \leq 2 \cdot \exp(-\frac{\mu\delta^2}{3}) \quad (29)$$

We know that  $\mu\delta = \frac{S}{N}$  and therefore  $\delta = \frac{S}{\mu N}$ :

$$P[|X - Sp| > \frac{S}{N}] \leq 2 \cdot \exp(-\frac{S^2}{\mu^2 N^2} \cdot \frac{\mu}{3}) \quad (30)$$

$$\leq 2 \cdot \exp(-\frac{S^2}{\mu N^2} \cdot \frac{1}{3}) \quad (31)$$

$$\leq 2 \cdot \exp(-\frac{S}{pN^2} \cdot \frac{1}{3}) \quad (32)$$

Since we want the original probability to be less or equal than  $\frac{1}{N}$ , we have to be sure that the bound found thanks to Chernoff is less than this value:

$$\frac{1}{N} \geq 2 \cdot \exp(-\frac{S}{pN^2} \cdot \frac{1}{3}) \quad (33)$$

$$\frac{1}{2N} \geq \exp(-\frac{S}{pN^2} \cdot \frac{1}{3}) \quad (34)$$

$$-\ln(2N) \geq -\frac{S}{3pN^2} \quad (35)$$

$$S \geq 3pN^2 \ln(2N) \quad (36)$$

Therefore, to get the desired “precision” in our estimate, we need to draw at least  $3pN^2 \ln(2N)$  samples from the random variable  $Y$ . Therefore, we can achieve the result with polynomially-many samples.

## Question 4

a)

Thanks to the linearity of the expectation,  $\mathbb{E}[X]$  is the sum of all the expectations of the indicators variables  $X_f$ . There are  $\binom{n}{4}$  possible  $X_f$  variables, given the original graph  $G$  (basically, they are equal to how many groups of 4 vertices we can form from  $G$ ):

$$\mathbb{E}[X] = \mathbb{E} \left[ \sum_{i=1}^{\binom{n}{4}} X_{f_i} \right] = \sum_{i=1}^{\binom{n}{4}} \mathbb{E}[X_{f_i}] \quad (37)$$

We have to compute the expectation of the single random variable  $X_f$ , which is the probability that its 4 vertices will generate a 4-cycle, without the “crossing edges”:

$$\mathbb{E}[X_f] = 3p^4(1-p)^2 \quad (38)$$

The factor 3 counts for all the possible arrangements of the four vertices (since it is a cycle composed by  $k$  vertices, we block one of them and then compute a permutation of the others  $(k-1)!$ . However, some of the permutations are symmetric with respect to the blocked vertex, and therefore there are just half of them.). The final formula for the expectation of  $X$  is:

$$\mathbb{E}[X] = \sum_{i=1}^{\binom{n}{4}} 3p^4(1-p)^2 = 3 \binom{n}{4} p^4(1-p)^2 \quad (39)$$

b)

Given that  $p = f(n)$  and  $\lim_{n \rightarrow \infty} f(n) = 0$  we can show that:

$$\lim_{n \rightarrow \infty} X_f = \lim_{n \rightarrow \infty} 3f(n)^4(1-f(n))^2 = 0 \quad (40)$$

This means that the probability that 4 vertices are arranged as a 4-cycle is equal to zero. Therefore, given the random variable  $X$ ,  $\lim_{n \rightarrow \infty} P[X > 0] = 0$ , because all the underlinig  $X_f$  which compose  $X$  will be equal to zero.