1. (a) $E(x) = \Sigma x \cdot P(x)$

where x = # of plants, $P(x)$ = probability of death

So we have total n # of plants from which 2 are on the either side and $(n-2)$ are in the mid.

So, probability of dying the side plant $= \frac{1}{4}$

$P(\text{side plant short}) \times P(\text{2nd plant high}) = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$

Now for mid plant probability of dying is

$= P(\text{1st side plant high}) \times P(\text{plant short}) \times P(\text{2nd side plant high})$

$= \frac{1}{8}$

Hence, expected number of dead plants

$= \Sigma x P(x) = 2 \times \frac{1}{4} + (n-2) \times \frac{1}{8}$

$\therefore \frac{n+2}{8}$

(b) There are $\frac{n(n-1)}{2}$ pair of students and each pair has a $\frac{1}{2}$ chance of leading to a complaint. So the expected number of complaints is

$E(n) = \frac{n(n-1)}{2} \times \frac{1}{2} = \frac{n(n-1)}{4}$

$\therefore \frac{n(n-1)}{4}$

2. Let X_i be a random variable that takes the value 1 if the subarray starting at position i and ending at position $i+k-1$ is increasing, and 0 otherwise.

$$E(\text{number of increasing subarrays of length } k) = \Sigma_{i=1}^{n-k+1} E(X_i)$$

Since the array is a random permutation, each permutation is equally likely, and there are $k!$ permutations where the subarray is in increasing order. There are $n-k+1$ possible starting positions, so: $E(X_i) = \frac{k!}{n-k+1}$

$$\therefore E(\#\text{ of increasing subarrays of length } k) = \Sigma_{i=1}^{n-k+1} \frac{k!}{n-k+1}$$

3. (a) (i) The algorithm returns

0 if $b_1 = 0$ and $b_2 = 1$

1 if $b_1 = 1$ and $b_2 = 0$

$P(\text{returning } 1) = p \cdot (1-p) + (1-p)p = 2p(1-p)$

$P(\text{returning } 0) = p(1-p) + (1-p)p = 2p(1-p)$

Since the probabilities of returning 0 and 1 are both $2p(1-p)$, the algorithm returns 0 and 1 with equal probability.

(ii) The probability of obtaining unbiased bit in one iteration is $(1-p)p + p(1-p) = 2p(1-p)$. The expected number of trials until the first success is $\frac{1}{p}$. In this case, the success is obtaining an unbiased bit, which happens with probability $2p(1-p)$ in one iteration. So, the expected number of iteration is $\frac{1}{2p-2p^2}$

(b) (i)

```
function Get-Unbiased-Die():
    loop
        result = Get-Biased-Die()  // Use Get-Biased-Die as a source of randomness

        // The following code maps the biased die result to an unbiased result
        // using rejection sampling.
        if result == 1 and Get-Biased-Bit() == 0:
            return 1
        else if result == 2 and Get-Biased-Bit() == 0:
            return 2
        else if result == 3 and Get-Biased-Bit() == 0:
            return 3
        else if result == 4 and Get-Biased-Bit() == 0:
            return 4
        else if result == 5 and Get-Biased-Bit() == 0:
            return 5
        else if result == 6 and Get-Biased-Bit() == 0:
            return 6
```

(II)

The provided algorithm ensures equal probability for each integer in {1, 2, 3, 4, 5, 6} through a careful application of rejection sampling. By using the biased die as a source of randomness and incorporating the biased bit from the Get-Unbiased-Bit algorithm, the algorithm accepts or rejects each outcome appropriately. The correctness of the rejection sampling process is established by checking for the conditions under which a biased die result is accepted, and the probability of accepting each outcome is proportional to the probability of obtaining that outcome from the biased die. The algorithm guarantees that the probability of acceptance for each integer is the same, namely 1/2, resulting in a uniform distribution over the set of possible outcomes and fulfilling the requirement of equal probability for each integer in the range {1, 2, 3, 4, 5, 6}.

(III)

To determine the expected number of calls that Get-Unbiased-Die makes to Get-Biased-Die, we can use the hint provided, which involves the geometric distribution. The geometric distribution gives the probability that the first occurrence of success (in this case, obtaining an unbiased result) requires k independent trials, each with success probability p. The formula for the expected number of trials until the first success is:

E[number of trials until the first success]=1/p

In this context, success is obtaining an unbiased result, and p is the probability of accepting a biased die outcome, which is 1/2 (as shown in the previous explanation). Therefore, the expected number of calls is 1/(1/2)=2.

So, in terms of the probability distribution of the six outcomes, the expected number of calls is 2.

4. (a) To prove, we need to show that no two vertices in S share an edge. The algorithm selects vertices based on random sampling, ensuring that for each vertex v, if $r_v > r_v'$ for every neighboring vertex v', then v is added to the set S. This selection process ensures that no two vertices in S share an edge, as if v is chosen, none of its neighbors with lower r values are included due to the random sampling condition.

(b) Let's denote (OPT) as the size of the maximum independent set in (G) and (|S|) as the size of the set returned by the algorithm. The probability that a vertex (v) gets selected into (S) is $(\frac{1}{\Delta+1})$.

Therefore, the expected size of (S) can be calculated as the sum of the probabilities each vertex being included:

$E[|S|] = \Sigma_{v \in V}$ Probability that v is in S

$E[|S|] = \Sigma_{v \in V} \frac{1}{\Delta+1}$

Now, the maximum independent set OPT can't have more than one vertex from each node, and the sum of the sizes of independent sets is at most $|V|$. Therefore, $OPT \leq |V|$.

$\therefore E[|S|] = \Sigma_{v \in V} \frac{1}{\Delta+1} \geq \frac{|V|}{\Delta+1} \geq \frac{OPT}{\Delta+1}$

The expected size of the output set S is within $(\frac{1}{\Delta+1})$ of (OPT).