Probabilistic Algorithms: Ex2

1. (a) The proabilistic **MAX-CUT** algorithm:
Input: A graph $G = (V, E)$
Output: A cut C such that $\mathbf{E}[Size(C)] \geq |E|/2$
$C = \emptyset$
For each $v \in V$
{
toss a fair coin.
if the result is HEADS put v in C.
}

Theorem 1 *The algorithm produces a cut C such that* $\mathbf{E}[Size(C)] \geq |E|/2$

Proof: For each edge $(u, v) \in E$ we'll define a random variable X_{uv} such that:

$$X_{uv} = \begin{cases} 1 & \text{if edge } (u,v) \text{ goes out from } C \\ 0 & \text{otherwise} \end{cases}$$

Thus we can write that $Size(C) = \sum_{(u,v) \in E} X_{uv}$, and we seek for $\mathbf{E}[Size(C)]$.

$$\mathbf{E}[Size(C)] = \mathbf{E}\left[\sum_{(u,v) \in E} X_{uv}\right]$$

Using the linearity of expectation we get

$$\mathbf{E}[Size(C)] = \sum_{(u,v) \in E} \mathbf{E}[X_{uv}]$$

Since X_{uv} is and indicator

$$\mathbf{E}[X_{uv}] = 1 \cdot \mathbf{Pr}[X_{uv} = 1] = \frac{1}{2}$$

We'll prove that $\mathbf{Pr}[X_{uv} = 1] = \frac{1}{2}$,keeping in mind that each vertex is chosen independently with equal probability to be in C or not.

$$\mathbf{Pr}[X_{uv} = 1] = \mathbf{Pr}[(u \in C \land v \notin C) \lor (u \notin C \land v \in C)] =$$

$$\mathbf{Pr}[u \in C] \cdot \mathbf{Pr}[v \notin C] + \mathbf{Pr}[u \notin C] \cdot \mathbf{Pr}[v \in C] = \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2}$$

Finally:

$$\mathbf{E}[Size(C)] = \sum_{(u,v) \in E} \mathbf{E}[X_{uv}] = |E|/2$$

■

(b) The derandomized version.
For the sake of clarity we'll assume that each vertex has an index $i (1 \le i \le n)$, meaning the vertices have some order.Therefore $V = \{v_1, v_2 \dots v_n\}$.
We also define

- $V_i = \{v_1 \dots v_i\}$, the set of vertices which have already been visited after the ith iteration.
- $C_i \subseteq V_i$, the set of vertices which have been chosen to be in C by the i iteration.
- $\overline{C_i} = V_i \backslash C_i$
- $\overline{V_i} = \{v_{i+1} \dots v_n\}$,

We can view the algorithm which creates a random cut C in the graph $G = (V, E)$ as a binary tree, where each node is a choice point and each path from the root to a leaf in the tree corresponds to a sequence of choices which determines a particular outcome of the random experiment. The root node corresponds to not having any choice yet, the vertices at level 1 corresponds to putting v_1 in C or not, and in general the nodes at level i corresponds to the ways of choosing the sides for V_i in the original graph.
Let $Y = \#$ of edges crossing the resulting cut .
We'll define $\mathbf{E}\big[Y | C_i, \overline{C_i}\big]$ to be the expected value of Y conditioned on the event that $C_i \subseteq C$ and $\overline{C_i} \cap C = \emptyset$.
We label each node at level i in the tree with the $\mathbf{E}\big[Y | C_i, \overline{C_i}\big]$. (Obviously for each node at level i there is a different sequence of choices made in order to reach it, meaning that for every node, C_i is a different set of vertices). According to the proof on previous section the root should be labelled with $|E|/2$, beacuse no choices have been done, yet. We call a node in the tree **good**, if its label $\mathbf{E}\big[Y | C_i, \overline{C_i}\big] \ge |E|/2$. Our goal is to find a path from the root to a leaf using only good nodes, and thus we'll reach $\mathbf{E}\big[Y | C_n, \overline{C_n}\big] \ge |E|/2$ meaning a cut in the original graph with the desired size.
We should notice that

$$\mathbf{E}\big[Y | C_{i-1}, \overline{C_{i-1}}\big] = \mathbf{Pr}[v_i \in C_i] \cdot \mathbf{E}\big[Y | C_{i-1} \cup \{v_i\}, \overline{C_{i-1}}\big] + \mathbf{Pr}\big[v_i \in \overline{C_i}\big] \cdot \mathbf{E}\big[Y | C_{i-1}, \overline{C_{i-1}} \cup \{v_i\}\big] =$$

$$\frac{1}{2} \cdot \mathbf{E}\big[Y | C_{i-1} \cup \{v_i\}, \overline{C_{i-1}}\big] + \frac{1}{2} \cdot \mathbf{E}\big[Y | C_{i-1}, \overline{C_{i-1}} \cup \{v_i\}\big]$$

From the above expression we can conclude that one of the following happens:

$$\mathbf{E}\big[Y | C_{i-1}, \overline{C_{i-1}}\big] \le \mathbf{E}\big[Y | C_{i-1} \cup \{v_i\}, \overline{C_{i-1}}\big]$$

or

$$\mathbf{E}\big[Y | C_{i-1}, \overline{C_{i-1}}\big] \le \mathbf{E}\big[Y | C_{i-1}, \overline{C_{i-1}} \cup \{v_i\}\big]$$

That means that every node has a child which is a good node.

In order to identify the good child, we have to evaluate $\mathbf{E}\big[Y|C_{i-1}, \overline{C_{i-1}} \cup \{v_i\}\big]$, and $\mathbf{E}\big[Y|C_{i-1} \cup \{v_i\}, \overline{C_{i-1}}\big]$.
To do that we define $\mathbf{Pr}\big[X_{v_s,v_t} = 1|C_i, \overline{C_i}\big]$ to be the probability of the edge (v_s, v_t) crossing the cut conditioned on the event of $C_i \subseteq C$.
$\mathbf{E}\big[Y|C_i, \overline{C_i}\big] = \sum_{(v_s,v_t)\in E} \mathbf{Pr}\big[X_{v_s,v_t} = 1|C_i, \overline{C_i}\big]$
The probability calculation over the edges is as follows:

$$\mathbf{Pr}\big[X_{v_s,v_t} = 1|C_i, \overline{C_i}\big] = \begin{cases} 1 & \text{if } v_s \in C_i, v_t \in \overline{C_i} \\ 1 & \text{if } v_s \in \overline{C_i}, v_t \in C_i \\ 0 & \text{if } v_s \in C_i, v_t \in C_i \\ 0 & \text{if } v_s \in \overline{C_i}, v_t \in \overline{C_i} \\ \frac{1}{2} & \text{if } v_s \in C_i, v_t \in \overline{V_i} \\ \frac{1}{2} & \text{if } v_s \in \overline{C_i}, v_t \in \overline{V_i} \\ \frac{1}{2} & \text{if } v_s \in \overline{V_i}, v_t \in C_i \\ \frac{1}{2} & \text{if } v_s \in \overline{V_i}, v_t \in \overline{C_i} \\ \frac{1}{2} & \text{if } v_s \in \overline{V_i}, v_t \in \overline{V_i} \end{cases}$$

The algorithm is as follows:
Input: A graph $G = (V, E)$
Output: A cut C_n such that $Size(C_n) \geq |E|/2$
$C_0 = \emptyset, \overline{C_0} = \emptyset$
For $i = 1$ to n
{

let $e1 = \mathbf{E}\big[Y|C_{i-1}, \overline{C_{i-1}} \cup \{v_i\}\big]$
let $e2 = \mathbf{E}\big[Y|C_{i-1} \cup \{v_i\}, \overline{C_{i-1}}\big]$
if $e2 \geq e1$
{
$\overline{C_i} = \overline{C_{i-1}}$
$C_i = C_{i-1} \cup \{v_i\}$
}
else
{
$\overline{C_i} = \overline{C_{i-1}} \cup \{v_i\}$
$C_i = C_{i-1}$
}
}

Theorem 2 *The algorithm produces a cut C such that $Size(C) \geq |E|/2$*

Proof: According to the detailed description above we get that
$\mathbf{E}\big[Y|C_n, \overline{C_n}\big] \geq \mathbf{E}\big[Y|C_{n-1}, \overline{C_{n-1}}\big] \geq \ldots \geq \mathbf{E}\big[Y|C_0, \overline{C_0}\big] \geq |E|/2$
This means that there is a cut C_n where $Size(C_n) \geq |E|/2$ ■

2. Let $i \in \{0,1\}^n$, i is of the form $a \circ b$,where a contains exactly k ones, and $b = 0^{n/2}$. Generally we're looking for a permutation σ such that $\sigma(a \circ b) = b \circ a$. That means that we have $2k$ bits that we have to change, and $|S| = 2k$. On a general random permutation π, the $\pi(\ell)$ bit is flipped on the ℓth step. Since there are $2k$ bits to be changed then there are $(2k)!$ possible random permutations σ.
Specifically we're looking for permutations which pass the packets through $0^{n/2} \circ 0^{n/2}$. To satisfy this restriction, all k ones of a must be changed to zeroes before the corresponding bits of b are changed to ones.

The number of sub-permutation which for step $1 \leq \ell \leq k$ $1 \leq \pi(\ell) \leq k$, is $k!$. The number of sub-permutations which for step $k < \ell \leq 2k$, $n/2 \leq \pi(\ell) \leq n/2 + k$, is $k!$.

Therefore, for a given node i of the form $a \circ 0^{n/2}$, the probability that the permutation $\sigma(a \circ 0^{n/2}) = 0^{n/2} \circ a$ will pass through $0^{n/2} \circ 0^{n/2}$ is:

$$\frac{k! \cdot k!}{(2k)!} = \frac{1}{\binom{2k}{k}}$$

We are intereseted with the expected number of such nodes i as described above. Since a contains k ones in its first $n/2$ bits ,then the number of such nodes is $\binom{n/2}{k}$.
For every node i of the form $a \circ 0^{n/2}$, where a contains exactly k ones, we'll define the random variable X_i such that:

$$X_i = \begin{cases} 1 & \text{if } \sigma(a \circ 0^{n/2}) = 0^{n/2} \circ a \text{ , and i sends its packet through } 0^{n/2} \circ 0^{n/2} \\ 0 & \text{otherwise} \end{cases}$$

Then the number of such nodes is $X = \sum_{i=1}^{\binom{n/2}{k}} X_i$.

$\mathbf{E}[X] = \mathbf{E}\left[\sum_{i=1}^{\binom{n/2}{k}} X_i\right] = \sum_{i=1}^{\binom{n/2}{k}} \mathbf{E}[X_i]$, by linearity of expectation.

Since X_i is indicator the $\mathbf{E}[X_i] = 1 \cdot \mathbf{Pr}[X_i = 1] = \frac{1}{\binom{2k}{k}}$. Therfore we can say:

$$\mathbf{E}[X] = \sum_{i=1}^{\binom{n/2}{k}} \mathbf{Pr}[X_i = 1] = \binom{n/2}{k} \cdot \frac{1}{\binom{2k}{k}} \geq \left(\frac{n/2}{k}\right)^k \cdot \frac{1}{(\frac{2k \cdot e}{k})^k} = \left(\frac{n}{4k \cdot e}\right)^k$$

Since $k = n/8e$ we can say that $\mathbf{E}[X] \geq \left(\frac{n}{4e \cdot n/8e}\right)^{n/8e} = 2^{\frac{n}{8e}} = 2^{\Omega(n)}$ That means that there exist a packet which needs $2^{\Omega(n)}$ steps to get to its destination. We want to know that this would happen with probabilty at least $3/4$.

Since all X_i are independenet 0/1 variables and $X = \sum_i X_i$ we can use Chernoff's inequality to determine how much X can get far from its expected value. According to Chernoff's inequality:

$$\mathbf{Pr}[X < (1-\delta)\mathbf{E}[X]] \leq 2^{-\frac{\delta^2 \mathbf{E}[X]}{6}}$$

If $\delta = \frac{1}{2}$ we would get the probability that half of the expected packets won't pass through $0^{n/2} \circ 0^{n/2}$.

$$\mathbf{Pr}\Big[X < 2^{\frac{n}{8e}}/2\Big] \leq 2^{-\frac{2^{\frac{n}{8e}}}{24}} \leq 1/4$$

And from this result we can conclude:

$$\mathbf{Pr}\Big[X \geq 2^{\frac{n}{8e}}/2\Big] = 1 - \mathbf{Pr}\Big[X < 2^{\frac{n}{8e}}/2\Big] \geq 1 - 2^{-\frac{2^{\frac{n}{8e}}}{24}} \geq 3/4$$

3. (a) Finding an independent set.

We'll set an order on the set of vertices, so that $V = \{v_1, v_2 \dots v_n\}$.

For each vertex $v_j (1 \leq j \leq n)$, we'll define its set of adjacent vertices , which determines its incident edges. $\hat{E}(v_j) = \{v_i | (v_i, vj) \in E \wedge (j > i)\}$

[IS] the independent-set algorithm:

Input: A graph $G = (V, E)$

Output: A set $S \subseteq V$ such for every $v_1, v_2 \in S$, v_1 and v_2 are not connected by an edge

$S = V$

for each $v \in S$

if $\hat{E}(v) \cap S \neq \emptyset$

$S = S \backslash \{v\}$ return S

Theorem 3 *The IS algorithm produces a independent set of size at least* $|V| - |E|$

Proof: By the construction of the algorithm we visit every $v_j \in V$ and remove it from S if there was some v_i $(i < j)$ such that $(v_i\, v_j)$ forms an edge. Removing v_j is actually removing the edges (v_i, v_j) (for $i < j$). By defining $\hat{E}(v_j)$ in the above way we ensure every edge to occur only once. That means that there is only one way to remove (v_i, v_j), $(i < j)$ which is removing v_j. We can remove at most $|E|$ vertices since there are $|E|$ edges and therefore we get an independent set S, where $|S| = |V| - |E|$. ■

(b) For each $v_i \in V$ we'll define a random variable X_i :

$$X_i = \begin{cases} 1 & \text{if } v_i \text{ is not deleted} \\ 0 & \text{otherwise} \end{cases}$$

Since the each vertex is deleted independently we can say that $\mathbf{Pr}[X_i = 1] = 1/d$. The number of remaining vertices is $X = \sum_{i=1}^{n} X_i$ and therefore $\mathbf{E}[X] = \mathbf{E}[\sum_{i=1}^{n} X_i]$. By linearity of expectation,we get $\mathbf{E}[X] = \sum_{i=1}^{n} \mathbf{E}[X_i] = \sum_{i=1}^{n} 1 \cdot \mathbf{Pr}[X_i = 1] = n/d$

For each $e_j \in E$ we'll define a random variable Y_j :

$$Y_j = \begin{cases} 1 & \text{if } e_j \text{ is not deleted} \\ 0 & \text{otherwise} \end{cases}$$

An edge e_j is not deleted iff both vertices which form it were not deleted. Therefore if $e_j = (v_k, v_m)$ then $\mathbf{Pr}[Y_j = 1] = \mathbf{Pr}[(v_k \text{ is not deleted}) \wedge (v_m \text{ is not deleted})] = \mathbf{Pr}[v_k \text{ is not deleted}] \cdot \mathbf{Pr}[v_m \text{ is not deleted}]$. The last equality is due to the independence in vertex deleting. Therefore $\mathbf{Pr}[Y_j = 1] = 1/d \cdot 1/d = 1/d^2$. The number of remaining edges is $Y = \sum_{j=1}^{nd/2} Y_j$ and therefore $\mathbf{E}[Y] = \mathbf{E}\left[\sum_{j=1}^{nd/2} Y_j\right]$. By linearity of expectation,we get $\mathbf{E}[Y] = \sum_{j=1}^{nd/2} \mathbf{E}[Y_j] = \sum_{j=1}^{nd/2} 1 \cdot \mathbf{Pr}[Y_j = 1] = \frac{nd}{2} \cdot \frac{1}{d^2} = \frac{n}{2d}$

(c) **[PIS]** the probabilisic independent-set algorithm:
Input: A graph $G = (V, E)$ $|V| = n, |E| = nd/2$ $(d > 1)$
Output: An independent-set $S \subseteq V$,$|S| = n/2d$
for each $v \in V'$
{
    roll a d sided die.
    if the result $\neq 1$ then
    {
        delete every $(v, w) \in E'$.
        delete v from V'.
    }
}
let $r = |V'|$
let $S = \{v_1, v_2 \dots v_r\}$.(each $v_i \in V'$ after the deletion phase and S sets order on the vertices of V')
For each vertex $v_j (1 \leq j \leq r)$, we'll define its set of adjacent vertices , which determines its incident edges. $\hat{E}(v_j) = \{v_i | (v_i, vj) \in E \wedge (j > i)\}$
for $j = 1$ to r
    if $\hat{E}(v_j) \cap S \neq \emptyset$
        $S = S \backslash \{v_j\}$

return S
On the first part of the algorithm (till the end of the first loop) we apply the algorithm from section 3(b), on the graph $G' = (V', E')$. When this part ends then according to the proof in 3(b) $|V'| = n/d$ and $|E'| = n/2d$.
On the second part of the algorithm, we apply the algorithm from section 3(a), on the graph $G'' = (V', E')$. This algorithm creates an independent set ,S, which expected size is (according to the proof on section 3(a)) at least $|V'| - |E'| = n/d - n/2d = n/2d$

(d) Let $V = \{v_1 \dots v_n\}$. We'll view the algorithm as a binary tree where in the ith step we choose whether to pick v_i to be in S or not , dependent on our sequnce of choices for $\{v_1 \dots v_{i-1}\}$.
We'll define the following.

- $V_i = \{v_1 \dots v_i\}$, the set of vertices which have already been visited after the ith iteration.
- $S_i \subseteq V_i$, the set of vertices which have been chosen to be in S by the i iteration.
- $\overline{S_i} = V_i \backslash S_i$
- $\overline{V_i} = \{v_{i+1} \dots v_n\}$,

In order to know the expected size of the independent set we have to calculate the difference between expected number of vertices and the expected number of edges in our sample.
Let $Z = \#$ of vertices remained in the independent set.

On section 3(b) we have definded:
X - number of vertices remained after the deleting method.
Y - number of edges remained after the deleting method.
Thus $\mathbf{E}[Z] = \mathbf{E}[X - Y]$. We'll define $\mathbf{E}\big[Z|S_i, \overline{S_i}\big]$,$\mathbf{E}\big[X|S_i, \overline{S_i}\big]$,$\mathbf{E}\big[Y|S_i, \overline{S_i}\big]$ to be the expected values of Z,X,Y respectively, conditioned on the event that $S_i \subseteq S$. $\mathbf{E}\big[Z|S_i, \overline{S_i}\big] = \mathbf{E}\big[X - Y|S_i, \overline{S_i}\big] = \mathbf{E}\big[X|S_i, \overline{S_i}\big] - \mathbf{E}\big[Y|S_i, \overline{S_i}\big]$
by linearity of expectation.

We label each node at level i in the tree with the $\mathbf{E}\big[Z|S_i, \overline{S_i}\big]$. According to the proof on previous section the root should be labelled with $n/2d$, beacuse no choices have been done, yet. We call a node in the tree **good**, if its label $\mathbf{E}\big[Z|S_i, \overline{S_i}\big] \geq n/2d$. We find a path from the root to a leaf using only good nodes, and thus we'll reach $\mathbf{E}\big[Z|S_n, \overline{S_n}\big] \geq$ meaning an independent-set in the original graph with the desired size.

$$\mathbf{E}\big[Z|S_{i-1}, \overline{S_{i-1}}\big] = \mathbf{Pr}[v_i \in S_i] \cdot \mathbf{E}\big[S|C_{i-1} \cup \{v_i\}, \overline{S_{i-1}}\big] + \mathbf{Pr}\big[v_i \in \overline{S_i}\big] \cdot \mathbf{E}\big[Z|C_{i-1}, \overline{S_{i-1}} \cup \{v_i\}\big] =$$

$$\frac{1}{d} \cdot \mathbf{E}\big[Z|S_{i-1} \cup \{v_i\}, \overline{S_{i-1}}\big] + (1 - \frac{1}{d}) \cdot \mathbf{E}\big[Z|S_{i-1}, \overline{S_{i-1}} \cup \{v_i\}\big]$$

From the above expression we can conclude that one of the following happens:

$$\mathbf{E}\big[Z|S_{i-1}, \overline{S_{i-1}}\big] \leq \mathbf{E}\big[Z|S_{i-1} \cup \{v_i\}, \overline{C_{i-1}}\big]$$

or

$$\mathbf{E}\big[Z|S_{i-1}, \overline{S_{i-1}}\big] \leq \mathbf{E}\big[Z|S_{i-1}, \overline{S_{i-1}} \cup \{v_i\}\big]$$

That means that every node has a child which is a good node.
In order to identify the good child, we have to evaluate $\mathbf{E}\big[Z|S_{i-1}, \overline{S_{i-1}} \cup \{v_i\}\big]$, and $\mathbf{E}\big[Z|S_{i-1} \cup \{v_i\}, \overline{S_{i-1}}\big]$.
That means we have to evaluate $\mathbf{E}\big[X|S_{i-1}, \overline{S_{i-1}} \cup \{v_i\}\big] - \mathbf{E}\big[Y|S_{i-1}, \overline{S_{i-1}} \cup \{v_i\}\big]$, and $\mathbf{E}\big[Y|S_{i-1}, \overline{S_{i-1}} \cup \{v_i\}\big] - \mathbf{E}\big[Y|S_{i-1} \cup \{v_i\}, \overline{S_{i-1}}\big]$.
To do that we define $\mathbf{Pr}\big[X_{v_k} = 1|S_i, \overline{S_i}\big]$ to be the probability of the vertex v_k wasn't deleted conditioned on the event of $S_i \subseteq S$.
we also define $\mathbf{Pr}\big[Y_{v_s,v_t} = 1|S_i, \overline{S_i}\big]$ to be the probability of the edge (v_s, v_t) wasn't deleted conditioned on the same event.
$\mathbf{E}\big[X|S_i, \overline{S_i}\big] = \sum_{k=1}^{n} \mathbf{Pr}\big[X_{v_k} = 1|S_i, \overline{S_i}\big]$
$\mathbf{E}\big[Y|S_i, \overline{S_i}\big] = \sum_{(v_s,v_t)\in E} \mathbf{Pr}\big[X_{v_s,v_t} = 1|S_i, \overline{S_i}\big]$
The probability over the vertices is as follows

$$\mathbf{Pr}\big[X_{v_k} = 1|S_i, \overline{S_i}\big] = \begin{cases} 1 & \text{if } v_k \in S_i \\ 0 & \text{if } v_k \in \overline{S_i} \\ \frac{1}{d} & \text{if } v_k \in \overline{V_i} \end{cases}$$

The probability calculation over the edges is as follows:

$$\mathbf{Pr}\big[Y_{v_s,v_t} = 1|S_i, \overline{S_i}\big] = \begin{cases} 1 & \text{if } v_s \in S_i, v_t \in S_i \\ 0 & \text{if } v_s \in S_i, v_t \in \overline{S_i} \\ 0 & \text{if } v_s \in \overline{S_i}, v_t \in \overline{S_i} \\ 0 & \text{if } v_s \in \overline{S_i}, v_t \in \overline{V_i} \\ \frac{1}{d} & \text{if } v_s \in S_i, v_t \in \overline{V_i} \\ \frac{1}{d^2} & \text{if } v_s \in \overline{V_i}, v_t \in \overline{V_i} \end{cases}$$

The algorithm is as follows:
Input: A graph $G = (V, E)$
Output: Independent-Set $S_n \in V$ such that $|S_n| \geq n/2d$
$S_0 = \emptyset, \overline{S_0} = \emptyset$
For $i = 1$ to n
{
    let $e1 = (1 - \frac{1}{d}) \cdot \mathbf{E}\big[Z|S_{i-1}, \overline{S_{i-1}} \cup \{v_i\}\big]$
    let $e2 = \frac{1}{d} \cdot \mathbf{E}\big[Z|S_{i-1} \cup \{v_i\}, \overline{S_{i-1}}\big]$
    if $e2 \geq e1$
    {
        $\overline{S_i} = \overline{S_{i-1}}$
        $S_i = S_{i-1} \cup \{v_i\}$
    }
    else
    {
        $\overline{S_i} = \overline{S_{i-1}} \cup \{v_i\}$
        $S_i = S_{i-1}$
    }
}

For each $v_j \in S_n, v_k \in S_n$ such that $k > j$

    if $(v_j, v_k) \in E$
        $S_n = S_n \backslash \{v_j\}$

Theorem 4 *The algorithm produces a cut S_n such that $S_n \geq n/2d$*

Proof: According to the description above we get that
$\mathbf{E}\big[Z|S_n, \overline{S_n}\big] \geq \mathbf{E}\big[Z|S_{n-1}, \overline{S_{n-1}}\big] \geq \ldots \geq \mathbf{E}\big[Z|S_0, \overline{S_0}\big] \geq n/2d$
Since $\mathbf{E}\big[Z|S_n, \overline{S_n}\big] = \mathbf{E}\big[X|S_n, \overline{S_n}\big] - \mathbf{E}\big[Y|S_n, \overline{S_n}\big]$ It means that the expected difference between number of vertices (X) and number of edges (Y) is at least $n/2d$, and the last part of the algorithm just deletes one vertex of each remaining edge. That means that we got ,deterministically, an independent-set S_n where $|S_n| \geq n/2d$ ■