An arbitrary starting assignment can disagree in all n variables. Then the bound from Lemma 4.3 is $2^{n+2} - 4 - 3n$. This is worse than the immediate upper bound of 2^n for (deterministically) iterating through all possible assignments to determine whether one of them is satisfying. We are interested in an algorithm with exponential running time, but the running time should be d^n for a $d < 2$.

Schöning's algorithm has two twists. First, the starting assignment is not chosen arbitrarily but uniformly at random from all possible starting assignments. Such an assignment will agree with a fixed a^* in $n/2$ variables on expectation. There is also a chance that it agrees in more variables. Second, Schöning's algorithm restarts the local search after a certain amount of steps. This is similar to the implicitly defined phases that we used in the Theorem 4.2, however, there is one important difference. After every restart, the algorithm again starts with an assignment chosen uniformly at random instead of the end assignment of the phase before. The following pseudo code `Improved-3-SAT`(ϕ, s) realizes Schöning's algorithm. It uses `Rand-3-SAT` as a subroutine. The parameter s controls the number of restarts. As in all of this section, n refers to the number of variables in ϕ in the pseudo code.

```
Improved-3-SAT(ϕ, s)
  1. for i = 1 to s do
  2.     choose an assignment a uniformly at random
  3.     if Rand-3-SAT(ϕ, 3n, a) = "YES" then return "YES";
  4. return "NO";
```

Let us first investigate whether starting with at least $n/2$ agreeing variables is a significant improvement over a worst case start (i. e. an assignment that disagrees in all variables). For $i = n/2$, the bound from Lemma 4.3 is $2^{n+2} - 2^{n/2+2} - 3(n/2)$ which is still $\Omega(2^{n+2})$. However, Lemma 4.3 only analyzes the *expected* number of steps that the random walk needs. In the previous sections, we often used statements on the expectation of a random variable and then argued that the random variable is close to this expected value because that was a good event for us. Here, we can also state that the number of steps will be close to $2^{n+2} - 2^{n/2+2} - 3(n/2)$ with constant probability. But we are now interested in the reverse question: Is there a good probability that the number of steps deviates significantly from its expected value, in particular, that it is significantly smaller?

We need a better understanding of the probability distribution, so we continue the analysis of the random walk. We want to find a lower bound for the probability that the walk reaches vertex n if it starts in vertex $n - i$. The probability space underlying this random process is rather complex. In particular, the walks are of different lengths. Assume we encode a walk starting in $n - i$ by a sequence of L and R that represent whether the algorithm made a step to the left or to the right. Then this sequence has at most t letters since the number of iterations is t. However, there are also walks that have fewer steps since the algorithm stops moving when it reaches n. The latter also means that not all strings of length t are feasible walks. For example, for $t > i$ the string $R^i(LR)^{(t-i)/2}$ does not represent a walk that can actually happen since the n is reached after i steps and then the algorithm stops making changes. Furthermore, a string that starts with L^{n-i+1} does not represent a walk because the algorithm would go to the right after reaching 0 after $n-i$ steps to the left.

Instead of analyzing this rather complex structure, we analyze a simplified random walk. We extend the line graph sufficiently to the left and right such that t consequtive steps in the same direction are possible from all vertices 0 to n. Furthermore, we assume that the walk acts in 0 and n as for all other nodes, having a probability of $\geq 1/3$ to go to right and a probability of $\leq 2/3$ to go to the left. Now we define the probability space Ω_i for $i \in \{0, \ldots, n\}$ as $\{L, R\}^{3i}$. This means that we consider all walks consisting of exactly $3i$ steps. (The number $3i$ will turn out to be a good choice later in our analysis). We define a *success* as the event that the walk ends in vertex n. In particular, if the walk visits n but the walk does not end in n, then this is counted as a failure. Observe that the probability for a success is a lower bound for the probability that we reach n from $n-i$ in the original random walk.

Let q_i be the probability of a success. In order to achieve a success, the walk has to end in vertex n after doing $3i$ steps. The walk reaches n by making k steps to the left and $i+k$ steps to the right, and in order to reach n in exactly $3i$ steps, k has to be i. Thus, every walk represented by a string with i L's and $2i$ R's leads to a success. There are $\binom{3i}{i}$ such strings in $\{L, R\}^{3i}$. Each of the corresponding walks has a probability of $(2/3)^i \cdot (1/3)^{2i}$. Thus, the probability for a success is

$$q_i = \binom{3i}{i} \cdot \left(\frac{2}{3}\right)^i \cdot \left(\frac{1}{3}\right)^{2i} = \binom{3i}{i} \cdot \frac{2^i}{3^{3i}} = \binom{3i}{i} \cdot \frac{2^i}{(3^3)^i} = \frac{(3i)!}{(2i)! \cdot i!} \cdot \frac{2^i}{27^i}. \tag{4.2}$$

Now we use Stirling's formula. This formula is due to Abraham de Moivre and James Stirling and states that $m!$ is approximately the same as $(m/e)^m$ times a smaller term, and that the smaller term is $\sqrt{2\pi m}$. For a proof of the following lemma, see for example 2.27 in [Mit70] (statement (6) in 2.27 gives tighter bounds from which the lemma follows).

Lemma 4.4 (Stirling's formula)**.** *It holds*

$$\sqrt{2\pi m} \cdot \left(\frac{m}{e}\right)^m \leq m! \leq 2\sqrt{2\pi m} \cdot \left(\frac{m}{e}\right)^m$$

for all $m \geq 1$.

By Lemma 4.4, $(3i)! \geq \sqrt{2\pi 3i} \cdot \left(\frac{3i}{e}\right)^{3i}$, $(2i)! \leq 2\sqrt{2\pi 2i} \cdot \left(\frac{2i}{e}\right)^{2i}$ and $i! \leq 2\sqrt{2\pi i} \cdot \left(\frac{i}{e}\right)^i$. We get that

$$\begin{aligned} q_i \geq \frac{(3i)!}{(2i)! \cdot i!} \cdot \frac{2^i}{27^i} &\geq \frac{\sqrt{2\pi 3i} \cdot (3i)^{3i}}{e^{3i}} \cdot \frac{e^{2i}}{2\sqrt{2\pi 2i} \cdot (2i)^{2i}} \cdot \frac{e^i}{2\sqrt{2\pi i} \cdot i^i} \cdot \frac{2^i}{27^i} \\ &= \frac{\sqrt{3}}{4\sqrt{4\pi}} \cdot \frac{1}{\sqrt{i}} \cdot \frac{(3i)^{3i}}{(2i)^{2i} i^i} \cdot \frac{2^i}{27^i} \\ &= \sqrt{\frac{3}{64\pi}} \cdot \frac{1}{\sqrt{i}} \cdot \left(\frac{(3i)^3 \cdot 2}{(2i)^2 \cdot i \cdot 27}\right)^i \end{aligned}$$

$$= \sqrt{\frac{3}{64\pi}} \cdot \frac{1}{\sqrt{i}} \cdot \left(\frac{27 \cdot 2}{4 \cdot 27}\right)^i = \frac{c}{\sqrt{i}} \cdot \frac{1}{2^i}$$

for $c = \sqrt{\frac{3}{64\pi}}$. In particular, $q_i \geq \frac{c}{\sqrt{n/2}} \cdot \frac{1}{2^{n/2}}$ for $i \leq n/2$. With this, we can show that `Improved-3-SAT` for an $s \in \mathcal{O}(1.42^n)$ with constant probability (exercise). This algorithm has a running time of $\mathcal{O}(n \cdot 1.42^n)$, an improvement over the 2^n bound for the brute force solution.

However, this result only used knowledge about reaching n from a vertex that is at least $n/2$. We can do better by using the full information that we gained about the q_i. Let q by the probability that one iteration of choosing an arbitrary assignment and doing $3n$ random walk steps successfully reaches vertex n, i.e. q is the success probability of one iteration of `Improved-3-SAT`. For any $i \in \{0, \ldots, n\}$, the probability to start in vertex $n-i$ is $p_i = \binom{n}{n-i}/2^n = \binom{n}{i}/2^n$. Let A_i be the event that the iteration starts in vertex $n - i$ and reaches n. The event A_i has probability $p_i \cdot q_i$. All events $A_i, i \in \{0, \ldots, n\}$ are disjoint, so $q = \sum_{i=0}^{n} \mathbf{Pr}(A_i) = \sum_{i=0}^{n} p_i \cdot q_i$. To compute a lower bound on this term, we use the binomial theorem.

Theorem 4.5 (Binomial theorem)**.** *For all $x, y \in \mathbb{R}$ and all $m \in \mathbb{N}_0$, it holds that*

$$(x+y)^m = \sum_{k=0}^{m} \binom{m}{k} \cdot x^{m-k} y^k.$$

Now we compute that

$$\begin{aligned} q &= \sum_{i=0}^{n} p_i \cdot q_i \\ &\geq \sum_{i=0}^{n} \frac{\binom{n}{i}}{2^n} \cdot \frac{c}{\sqrt{i}} \cdot \frac{1}{2^i} \geq \sum_{i=0}^{n} \frac{\binom{n}{i}}{2^n} \cdot \frac{c}{\sqrt{n}} \cdot \frac{1}{2^i} \\ &= \frac{c}{\sqrt{n}2^n} \sum_{i=0}^{n} \binom{n}{i} \cdot \left(\frac{1}{2}\right)^i \cdot 1^{n-i} \\ &= \frac{c}{\sqrt{n}2^n} \left(\frac{3}{2}\right)^n = \frac{c}{\sqrt{n}} \cdot \left(\frac{3}{4}\right)^n. \end{aligned} \tag{4.3}$$

We have now gathered enough information to prove the main theorem of this section.

Theorem 4.6. *Let ϕ be a satisfiable 3-SAT formula. For $s = \lceil \frac{(\ln 1/\delta) \cdot \sqrt{64\pi \cdot n}}{\sqrt{3}} \cdot (\frac{4}{3})^n \rceil$, the algorithm **`Improved-3-SAT`**$(\phi, s)$ finds a satisfying assignment for ϕ with probability $1 - \frac{1}{n}$ and has a running time of $\Theta((\ln 1/\delta) \cdot n^{3/2} \cdot 1.334^n)$.*

Proof. Recall that $c = \sqrt{3}/\sqrt{64\pi}$. By (4.3), the success probability of one iteration of `Improved-3-SAT`(ϕ, s) is at least $q' = \frac{c}{\sqrt{n}} \cdot (3/4)^n$. The probability that all s tries fail is at most $(1 - q')^s \leq (1 - q')^{(\ln 1/\delta)/q'} \leq (1/n)$. The running time for one iteration is $\Theta(n)$, so the total running time is $\theta(s \cdot n)$. □