# Testing the Diameter of Graphs

Michal Parnas
The Academic College
of Tel-Aviv-Yaffo
Tel-Aviv, ISRAEL
michalp@server.mta.ac.il

Dana Ron
Department of EE – Systems
Tel-Aviv University
Ramat Aviv, ISRAEL
danar@eng.tau.ac.il

**Abstract**

We propose a general model for testing graph properties, which extends and simplifies the bounded degree model of [GR97]. In this model we present a family of algorithms that test whether the diameter of a graph is bounded by a given parameter $D$, or is $\epsilon$-far from any graph with diameter at most $\beta(D)$. The function $\beta(D)$ ranges between $D+4$ and $4D+2$, depending on the algorithm. All our algorithms run in time polynomial in $1/\epsilon$.

**Keywords:** Graph Algorithms, Approximation Algorithms, Property Testing.

# 1 Introduction

*Testing Graph Properties* [GGR98] is the study of the following family of tasks. Let $P$ be a predetermined graph property (such as connectivity or 3-colorability), and let $G = (V, E)$ be a graph where $|V| = n$. The goal of the testing algorithm is to decide whether the graph $G$ has property $P$, or whether it differs significantly from any graph having the property. In other words, the algorithm should accept every graph that has the property, and reject every graph for which many edge modifications should be performed so that the graph has the property. To this end the algorithm is given *access* to the graph in the form of being able to *query* on the incidence relationship between vertices. The testing algorithm is allowed a constant probability of error, and should perform its task by observing as few vertices and edges in the graph as possible.

Thus, testing is a relaxation of *exactly deciding* whether the graph has the property, and testing algorithms may be useful in several contexts. In particular, if the graph considered is very large, then we may not have time to even scan all the graph in order to make the exact decision. In this case we must make an approximate decision based on seeing only a small (possibly random) part of the graph. Even if the graph is not very large, we may prefer a very fast (sub-linear in the size of the graph) approximate procedure to an exact procedure that runs in time linear or super-linear in the size of the graph.

In order to formulate testing graph properties more precisely, we need to answer two questions: (1) How are graphs represented, and what form of access to the graph is the testing algorithm given? (2) How do we measure the difference or distance between graphs (and what do we mean by "many" edge modifications)?

**Adjacency-Matrix Model.** Goldreich et. al. [GGR98] considered the adjacency matrix representation of graphs, where the testing algorithm is allowed to probe into the matrix. That is, the algorithm can query whether there is an edge between any two vertices of its choice. They define the distance between graphs as the fraction of entries in the adjacency matrix on which the two graphs differ. By this definition, for a given distance parameter $\epsilon$, the algorithm should reject every graph that requires more than $\epsilon \cdot n^2$ edge modifications in order to acquire the tested property. Hence, the notion of distance is directly determined by the representation of the graphs.

This representation is most appropriate for dense graphs, and the results for testing in this model are most meaningful for such graphs. However, if we are interested in testing properties of sparse graphs, then the above model might not be suitable. For example, if a graph has $o(n^2)$ edges then it is trivially close (for any constant distance) to having every property that the empty graph has.

**Bounded-length Incidence-Lists Model.** This prompted the consideration of a different representation of graphs which can be used when testing properties of sparse graphs. A natural alternative is that based on *incidence lists*, which is studied in [GR97]. In [GR97], graphs are represented by incidence lists of *length* $d$, where $d$ is a bound on the degree of the graph. Here the testing algorithm can query, for every vertex $v$ and index $i \in \{1, \ldots, d\}$, who is the $i$'th neighbor of $v$. Analogously to the adjacency matrix model, the distance between graphs is defined to be the fraction of entries on which the graphs differ according to this representation. Since the total number of incidence-list entries is $d \cdot n$, a graph should be rejected if the number of edges modifications required in order to obtain the property is greater than $\epsilon \cdot dn$.

This model is most appropriate for testing graphs that not only have an upper bound $d$ on their degree, but for which the number of edges is $\Omega(d \cdot n)$. In particular this model is suitable

for testing graphs that have a constant bound on their degree. However, if we want to study non-constant degree bounds then we may run into a difficulty similar to the one that arose in the adjacency matrix model. Namely, a graph may be considered close to having a certain property simply because it has relatively few edges with respect to $d \cdot n$, and so it is close to the empty graph. In particular, this problem arises when there is a large variance between the degrees of vertices in the graph.

**Functional Representations.** The above two models were defined to fit into the framework of *Property Testing of functions* [RS96, GGR98]. That is, the two graph representations described above correspond to *functional* representation of graphs. In the first case these are functions from pairs of vertices to $\{0,1\}$. In the second case these are functions from a vertex and an index in $\{1,\ldots,d\}$ to another vertex (or 0 if the neighbor does not exist). In both cases, the distance between graphs is determined by the representation. Namely, it is the distance between the corresponding functions, which is simply the fraction of domain-elements on which the two functions differ. The results of testing are most meaningful, if the number of edges in the tested graph is of the order of the size of the domain of the function. However, for the general case, we may need to seek a different representation, and an alternative distance measure.

### A Non-Functional Model for Testing Graph Properties

In this paper we propose a general model which is appropriate for testing properties of all (non-dense) graphs. To this end we have chosen to divorce ourselves from the functional representation of graphs. Furthermore, the question of distance measure is treated separately from the question of representation. In this model graphs are represented by incidence lists of *varying lengths*. For each vertex $v$, the testing algorithm can obtain the degree of $v$, denoted $d(v)$, and can query, for every $i \leq d(v)$, who is the $i$'th neighbor of $v$. Thus, there is no fixed functional relationship (and in particular, no fixed domain). The distance between graphs is defined independently of the representation. It is measured by the fraction of edge modifications that should be performed in order to obtain the property, where this fraction is defined *with respect to the number of graph edges*. More precisely, since we do not want to assume that the algorithm is given the exact number of edges $|E|$ in the graph, we assume it is given an upper bound $m$ on this number. For concreteness, we may think of $m$ as being $O(|E|)$. For any given $\epsilon$, a graph is said to be $\epsilon$-*far* from having the property (and should be rejected), if the number of edge modifications performed so that the graph obtains the property is greater than $\epsilon \cdot m$. The implicit assumption is that $m$ is in fact not much larger than the actual number of edges in the graph (but does not necessarily equal to this number).

By considering this generalization, we "open the door" to studying properties of graphs whose vertices vary significantly in their degrees. Furthermore, some problems are more interesting in this model, in the sense that removing the degree bound makes them less restricted. For example, the Diameter Problem presented in this paper, is less interesting in the bounded degree model, since a bound $d$ on the degree implies a lower bound on the diameter of a graph. Another example is testing whether a graph has a clique of a given size $k$ (or is far from any graph having a clique of size $k$). This question is meaningless in the bounded degree model for any $k > d+1$.

In our testing model we also allow for the following relaxation of the testing task (which was previously studied in [KR98]). Consider a parameterized property $P = \{P_s\}$, where $P_s$ may be for example: "Having diameter at most $s$". We assume the property is such that for every $s' \geq s$, if a graph has property $P_s$ then it has property $P_{s'}$. For any given parameter $s$, the algorithm

is required: (1) to accept every graph with property $P_s$; (2) to reject every graph that is $\epsilon$-far from having property $P_{\beta(s)}$, where $\beta(\cdot)$ is a certain *boundary function* such that $\beta(s) \geq s$. Thus, while in the previous definition of testing, $\beta(\cdot)$ was always the identity function, here we allow for different functions. However, we strive to obtain results where $\beta(s)$ is as close as possible to $s$. This relaxation is suitable whenever it is sufficient to know that a graph is close to having property $P_{\beta(s)}$. This seems to be true of many parameterized properties.

### Testing the Diameter of Graphs

We consider the problem of testing whether the diameter of a graph is within a certain upper bound. Our main result is a family of algorithms that determine (with probability at least 2/3), whether the diameter of a graph is bounded by a given parameter $D$, or is $\epsilon$-far from any graph with diameter at most $\beta(D)$. The function $\beta(D)$ is a small linear function, which ranges between $D + O(1)$ and $O(D)$, depending on the algorithm. In particular:

- For $\beta(D) = 2D + 2$, we have an algorithm that works for every $\epsilon$ and has one-sided error (that is, it always accepts a graph that has diameter at most $D$).
- For every $\beta(D) = (1 + \frac{1}{2^i - 1}) \cdot D + 2$ (where $i > 1$), we have a testing algorithm that has two-sided error, and works for $\epsilon$ that is lower bounded as a function of $i$, $n$, and $m$ (where $n = |V|$). For example, when $i = 2$, we obtain $\beta(D) = \frac{4}{3}D + 2$, and the algorithm works for $\epsilon = \Omega\left(\frac{n^{3/4} \log n}{m}\right)$, which in the worst case of $m = O(n)$ implies $\epsilon = \tilde{\Omega}(n^{-1/4})$.

All algorithms have query complexity and running time $\tilde{O}(\epsilon^{-3})$. In fact, the complexity improves as the number of edges $m$ grows compared to $n$. Specifically, the complexity is bounded by $\tilde{O}\left(\left(\frac{m}{n} \cdot \epsilon\right)^{-3}\right)$.

In view of our positive results it is interesting to note that the problem of finding the minimum number of edges that should be added to a graph in order to transform it into a graph with diameter at most $D$ is NP-hard [LMSL92]. Furthermore, in [LMSL92] evidence is given to the difficulty of approximating this minimum number.

### Transforming results from the bounded-degree model

Many of the results proved in [GR97] (for the bounded degree model) can be transformed smoothly to our general model. In particular, the algorithms for $k$-connectivity (as well as for testing whether a graph is Eulerian), essentially work as is. While we need to "pay" an additional factor of $1/\epsilon$ in their implementation (due to the removal of the degree bound), parts of the analysis are simplified for the same reason. Namely, when proving in [GR97], that every graph which is accepted is close to being $k$ connected, it is actually required that the graph be close to a $k$-connected graph with degree bounded by $d$. This adds technical difficulties that are avoided in our model. In addition, the hardness results proved there apply with no change to our model. In fact some of these lower bounds are much easier to prove in the general model (i.e., Bipartiteness). The algorithms that cannot be transformed to our model, either have an explicit dependency on $d$ (Planarity), or an implicit dependency (Cycle-Freeness). Among other things, the Cycle-Freeness testing algorithm approximates the number of edges in a subgraph of the tested graph. This cannot be performed efficiently (for the desired accuracy) when the degrees are unbounded. Furthermore, it can be shown that *any* algorithm for Cycle-Freeness in the general model requires $\Omega(\sqrt{|V|})$ queries (while in the bounded degree model there was no dependence on $|V|$).

**Organization**

The remainder of the paper is structured as follows. In Section 2 we present the general model. In Section 3 we define the diameter problem and present a family of testing algorithms for this problem. In Section 4 we prove a series of lemmas in which we show how to reduce the diameter of a graph by adding a small number of edges. Building on these lemmas, we prove in Section 5, the main theorem which states the performance and correctness of the testing algorithms. In Section 6 we discuss the transformations of results from the bounded-degree model to our model.

# 2 Model Definition

Let $G = (V, E)$ be an undirected graph where $|V| = n$. We represent graphs by *incidence lists* of possibly varying lengths, where the length of each list (i.e., the degree of the vertex the list corresponds to) is provided at the head of the list. For any vertex $v$, let $d(v)$ denote the degree of $v$ in the graph $G$.

Let $P = \{P_s\}$ be a *parameterized* graph property (e.g., $P_s$ may be the property of having diameter at most $s$).

**Definition 2.1** *Let $P_s$ be a fixed parameterized property, $0 < \epsilon < 1$, and $m$ a positive integer. A graph $G$ having at most $m$ edges is* $\epsilon$-far *from property $P_s$ (with respect to the bound $m$), if the number of edges that need to be added and/or removed from $G$ in order to obtain a graph having property $P_s$, is greater than $\epsilon \cdot m$. Otherwise, $G$ is* $\epsilon$-close *to $P_s$.*

A *testing algorithm* for (parameterized) property $P_s$, with boundary function $\beta(\cdot)$, is given a (size) parameter $s > 0$, a distance parameter $0 < \epsilon < 1$, a bound $m > 0$, and query access to an unknown graph $G$ having at most $m$ edges. Namely, in accordance with the above representation of graphs, the algorithm can query, for any vertex $v$ and index $1 \leq i \leq d(v)$, what is the $i$'th vertex incident to $v$.[1] The output of the algorithm is either accept or reject. We require that:

1. If $G$ has property $P_s$, then the algorithm should output accept with probability at least $2/3$;
2. If $G$ is $\epsilon$-far from property $P_{\beta(s)}$, then the algorithm should output reject with probability at least $2/3$.

We shall be interested in bounding the query complexity and running time of testing algorithms as a function of the distance parameter $\epsilon$, and possibly the size parameter $s$.
For an illustration of the testing task see Figure 1.

# 3 The Diameter Testing Problem

We present a family of algorithms that test whether the diameter of a graph is bounded by a given parameter $D$. The algorithms differ in the following aspects: (1) The boundary function $\beta(\cdot)$; (2) The query and time complexities; (3) The values of $\epsilon$ for which they can be applied. For brevity of the presentation we shall think of $m$ as being the actual number of edges in the graph.

We first establish that if $\epsilon$ is above some threshold (dependent on $D$, $n$ and $m$), then every *connected* graph with $n$ vertices and $m$ edges is $\epsilon$-close to having diameter $D$. Therefore, for these

[1] Thus it is implicitly assumed that $n = |V(G)|$ is known (at least to within a constant factor of 2) just so that the testing algorithm can refer to each vertex by its $\lceil \log n \rceil$-bits name.

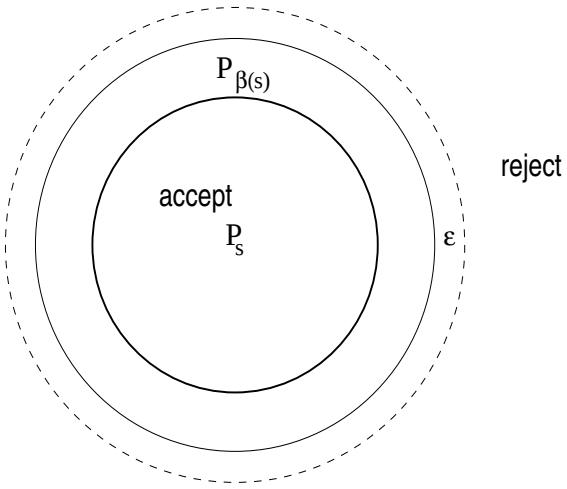


Figure 1: An illustration of the testing task.

values of $\epsilon$ (or more precisely, if $\epsilon$ is at least twice this lower bound) the testing algorithm can perform a connectivity test [GR97] (with distance parameter $\frac{\epsilon}{2}$), and accept if this test passes. Hence our interest lies mainly in smaller values of $\epsilon$. Note that we may also assume that $m \geq n-1$ or otherwise we know that the graph is disconnected and may reject it without testing. The following theorem, whose proof is omitted, was proved independently by Alon, Gyárfás and Ruszinkó [AGR99].

**Theorem 3.1** *Every connected graph on $n$ vertices can be transformed into a graph of diameter at most $D$ by adding at most $\frac{n}{\lfloor D/2 \rfloor}$ edges.*

As an immediate corollary we get:

**Corollary 3.2** *Every connected graph with $n$ vertices and $m$ edges is $\epsilon$-close to having diameter $D$ for every $\epsilon \geq \frac{2}{D} \cdot \frac{n}{m}$.*

Having established what is the "interesting" range of $\epsilon$, we next state our main theorem. From now on we shall assume that in fact $\epsilon < \frac{2}{D} \cdot \frac{n}{m}$. For any given $n$, $m$, and $\epsilon$, define

$$\epsilon_{n,m} \stackrel{\text{def}}{=} \frac{m}{n} \cdot \epsilon \ . \tag{1}$$

Recall that we assume that $m \geq n-1$, and so, with the exception of the case $m = n-1$, we have $\epsilon_{n,m} \geq \epsilon$. Notice also, that for any fixed $n$, as $m$ grows and the graph is denser, $\epsilon_{n,m}$ increases.

**Theorem 3.3**

1. *There exists a testing algorithm for diameter $D$ with boundary function $\beta(D) = 4D+2$, whose query and time complexities are $O(1/\epsilon_{n,m}^3)$. Furthermore, the testing algorithm has 1-sided error, that is, it* always *accepts graphs with diameter at most $D$.*

2. *There exists a testing algorithm for diameter $D$ with boundary function $\beta(D) = 2D+2$, whose query and time complexities are $O(\frac{1}{\epsilon_{n,m}^3} \cdot \log^2(1/\epsilon_{n,m}))$, and which has 1-sided error.*

3. *For any integer $2 \leq i \leq \log(D/2+1)$, there exists a testing algorithm for diameter $D$ with boundary function $\beta(D) = D(1+\frac{1}{2^i-1})+2$, whose query and time complexities are $O(\frac{1}{\epsilon_{n,m}^3} \cdot \log^2(1/\epsilon_{n,m}))$, and which works for every $\epsilon = \Omega\left(\frac{n^{1-\frac{1}{i+2}} \cdot \log n}{(i+2) \cdot m}\right)$.*

We first consider the application of Item 3 for different settings of $i$. Note that the lower bound on $\epsilon$ translates to $\epsilon_{n,m} = \Omega\left(n^{-\frac{1}{2+i}} \cdot \frac{\log n}{i+2}\right)$, and we shall find it more convenient to refer to the latter bound. If $i = 2$ then Item 3 implies that we can distinguish between graphs with diameter $D$ and graphs which are $\epsilon$-far from diameter $\frac{4}{3}D + 2$ for every $\epsilon_{n,m} = \Omega(n^{-1/4} \cdot \log n)$. As we increase $i$, our boundary function $\beta(D)$ gets closer and closer to $D$, while the lower bound on $\epsilon_{n,m}$ becomes a larger inverse root of $n$. At the extreme setting of $i = \log(D/2+1)$, we get $\beta(D) = D + 4$, and $\epsilon_{n,m} = \Omega\left(n^{-\frac{1}{\log(D/2)+2}} \cdot \frac{\log n}{\log D}\right)$. Hence for very large $D$ (say $D = n^{\alpha}$ for some $\alpha \leq 1$), this result is applicable for constant $\epsilon_{n,m}$, while for $D = \mathrm{poly}(\log n)$, it is applicable for $\epsilon_{n,m} = n^{-\Omega\left(\frac{1}{\log\log n}\right)}$ (and in particular for $\epsilon_{n,m} = 1/\mathrm{poly}(\log n)$).

We now describe the testing algorithm on which the theorem is based. Define the *$C$-neighborhood* of a vertex $v$ to be the set of all vertices at distance at most $C$ from $v$ (including $v$ itself), and denote it by

$$\Gamma_C(v) = \{u \mid \mathrm{dist}(u,v) \leq C\}$$

where $\mathrm{dist}(u,v)$ denotes the distance between $u$ and $v$ in $G$. The basic testing algorithm, which is given in Figure 2, uniformly and independently selects a sample of $O(1/\epsilon_{n,m})$ vertices, and checks whether the $C$-neighborhood of all but an $\alpha$-fraction of these vertices contains at least $k$ vertices. Since in the worst case, each such search may traverse $\Theta(k^2)$ edges, the query and time complexities of this algorithm are $O(k^2/\epsilon_{n,m})$. The testing algorithms referred to by Theorem 3.3, are essentially variants of this algorithm, and they differ only in the settings of the parameters $C$, $k$ and $\alpha$.

**Algorithm Test-Diameter**

- Given $D$, $n$, $m$ and $\epsilon$ as input, let $C$, $k$, and $\alpha$ be set as functions of these parameters (where the particular setting depends on the variant of the algorithm).
- If $k = \Omega(n)$ then determine whether the graph has diameter at most $D$ or not by observing the whole graph. Otherwise:
  1. Uniformly and independently select $S = \Theta(\frac{1}{\epsilon_{n,m}})$ starting vertices where $\epsilon_{n,m}$ is as defined in Equation (1).
  2. For each vertex selected, perform a Breadth First Search until $k$ vertices are reached, or all vertices at distance at most $C$ have been reached (whichever occurs first).
  3. If the fraction of starting vertices that reach distance $C$ before observing $k$ vertices (i.e., their $C$-neighborhood is of size less than $k$) is at most $\alpha$, then accept, otherwise reject.

Figure 2: The diameter testing algorithm.

**Size of Diameter vs. Distance to Diameter $D$** Before we start analyzing the algorithm, we pause for a short discussion concerning the relation between the size of the diameter and the distance to having a certain diameter.

One may ask the question whether being close to a graph with diameter not much larger than $D$ (e.g., diameter $D + 2$) implies something about the distance to diameter $D$. Clearly, there are graphs that have diameter $D+2$ and are very close to having diameter $D$ (e.g., a graph that consists of a clique with a "tail" of length $D + 1$ can be transformed into a graph with diameter at most $D$

by adding a single edge). However, as we see in the next theorem, there are such graphs that are quite far from diameter $D$.

**Theorem 3.4** *There exist graphs that have diameter $D+2$, that are $\Omega(1/D)$-far from having diameter $D$. (Recall that by Corollary 3.2, for $D > 2$, every connected graph is $O(1/D)$-close to having diameter $D$.)*

**Proof:** Consider the *star* graph that has $\frac{n-1}{D/2+1} \approx 2n/D$ "rays" of length $D/2+1$. This graph has diameter $2 \cdot (D/2+1) = D+2$. We claim that in order to transform this graph into a graph of diameter $D$, we must add at least one edge to some vertex on every ray but at most one. Assume there are two rays to which we did not add any edge. Then no matter what other edges are added, the distance between the two vertices at the end of these rays is $D+2$. It follows that this graph is $\Omega(1/D)$-far from diameter $D$, as claimed. □

# 4 Reducing the Diameter of a Graph

Our goal is to reduce the diameter of graphs for which the $C$-neighborhoods of all but possibly a small fraction of the vertices, contain $k$ vertices. The main technique we use to obtain a diameter of some bounded size is to select a set of $R$ vertices as "representatives", such that each vertex in the graph is at distance at most $\ell$ from some representative. We then connect all representatives to one designated representative $v_0$. Thus, the number of edges added to the graph is $R-1$. The distance between any two vertices $u, w$ in the new graph is at most $2\ell+2$, because $u$ is at distance at most $\ell$ from some representative $v_u$, $w$ is at distance at most $\ell$ from some representative $v_w$, and the distance between $v_u$ and $v_w$ is at most 2 since both are connected to $v_0$. Our technique is reminiscent of those applied in [DF85, Awe85, AP90, LS93].

**Remark:** By adding an edge between every representative and the designated representative $v_0$, we may be increasing the degree of that vertex significantly. In some cases it may be desired to bound the increase in the degree of the vertices in the graph. To this end we can connect the representatives so that they constitute nodes of a bounded degree *tree*.

The following results differ in the way the representatives are selected, and thus exhibit a tradeoff between the number $R$ of representatives and the distance $\ell$ of each vertex to its nearest representative.

**Lemma 4.1** *If the $C$-neighborhood of each vertex contains at least $k$ vertices, then the graph can be transformed into a graph with diameter at most $4C+2$ by adding at most $\frac{1}{k} \cdot n$ edges.*

**Proof:** For any vertex $v$, we shall view the set of vertices in the $C$-neighborhood of $v$ as a *ball of radius $C$*, centered at $v$. We shall say that a vertex $u$ is in the *$C$-boundary* of the ball if it is at distance at most $C$ from some vertex in the ball (i.e., it is at distance at most $2C$ from the center). We partially cover the graph with (disjoint) balls of radius $C$ in the following iterative manner: The center of the first ball can be any vertex. At any subsequent step, the next center selected can be any vertex not contained in any of the previous balls nor in their $C$-boundaries. Thus, every new ball is disjoint from all previous balls. When no such new center can be found we know that every vertex is at distance at most $2C$ from some previously selected center. Furthermore, the number of centers is at most $\frac{1}{k} \cdot n$.

Now connect the centers of all balls to the center of the first ball. The number of edges added is at most $\frac{1}{k} \cdot n - 1$, and the diameter of the resulting graph is at most $4C + 2$, since every vertex is at distance at most $2C$ from some center. $\square$

As an immediate corollary we get:

**Corollary 4.2** *For any $D \geq 2$, every connected graph on $n$ vertices can be transformed into a graph of diameter at most $D$, by adding at most $\frac{4}{D-1} \cdot n$ edges.*

**Proof:** The proof follows directly from the above lemma (by setting $C = \lfloor \frac{D-2}{4} \rfloor$, and $k = C+1$). $\square$

As stated before we can actually prove a slightly stronger result than the result stated in the last corollary (i.e., Theorem 3.1).

In the next lemma we slightly modify the premise of Lemma 4.1 so as to allow a small fraction of vertices whose $C$-neighborhoods contain less than $k$ vertices.

**Lemma 4.3** *If the $C$-neighborhood of at least $(1 - \frac{1}{k})n$ of the vertices contains at least $k$ vertices, then the graph can be transformed into a graph with diameter at most $4C + 2$ by adding at most $\frac{2}{k} \cdot n$ edges.*

**Proof:** The proof follows the same lines as the proof of Lemma 4.1, except here we only allow the vertices that have at least $k$ vertices in their $C$-neighborhood to be centers of balls of radius $C$. The number of balls is at most $\frac{1}{k} \cdot n$. Now connect the centers of all balls to one designated center, and connect the remaining (at most $\frac{1}{k} \cdot n$) vertices that are at distance greater than $2C$ from any center, to the same center. The total number of edges added is at most $\frac{2}{k} \cdot n$. $\square$

In the following Lemma we improve (by a factor of 2) the size of the diameter we can achieve, at a cost of a small increase in the required size of the $C$-neighborhood of most vertices. This is done by ensuring that the representatives selected be at distance at most $C$ from every vertex (as opposed to $2C$ in the previous lemma), while allowing their $C$-neighborhoods to overlap.

**Lemma 4.4** *If the $C$-neighborhood of at least $(1-\frac{\delta}{2})n$ of the vertices contains at least $k = \frac{4}{\delta} \ln(4/\delta)$ vertices for some $\delta < 1$, then the graph can be transformed into a graph with diameter at most $2C+2$ by adding at most $\delta n$ edges.*

In order to prove the lemma we shall need the following Claim.

**Claim 4.5** *Let $S_1, \ldots, S_Y$ be sets over $\{1, \ldots, n\}$, such that $Y \leq n$, and each set contains at least $X \geq \frac{2}{\delta} \ln(2/\delta)$ elements for some $\delta < 1$. Then there exists a* blocking set *$T \subset \{1, \ldots, n\}$ of $\{S_1, \ldots, S_Y\}$, with at most $\delta n$ elements. Namely, $T$ is such that for every $i \in \{1, \ldots, Y\}$, we have $S_i \cap T \neq \emptyset$.*

**Proof:** The proof is very similar to that given in [AS92] for bounding the size of a dominating set in a graph whose minimum degree is bounded below.

Consider the following greedy selection process.

- Starting from $T = \emptyset$, and $\mathcal{S} = \{S_1, \ldots, S_Y\}$, and until $|\mathcal{S}| \leq \frac{\delta}{2} n$, do the following:
  1. Select an element $x \in \{1, \ldots, n\}$ that belongs to the largest number of sets in $\mathcal{S}$.
  2. Add $x$ to $T$ and remove from $\mathcal{S}$ all sets which $x$ belongs to.

- When $|\mathcal{S}| \leq \frac{\delta}{2}n$, arbitrarily select one element from each set in $\mathcal{S}$ and add it to $T$.

In order to show that $|T| \leq \delta n$, it suffices to show that the number of elements added to $T$ in the greedy stage of the process is at most $\frac{\delta}{2}n$. Details follow.

Let $Y_i$ be the number of sets in $\mathcal{S}$ at the start of step $i$ where $Y_0 = Y \leq n$. Since the sets are all of size at least $X$, while the number of elements in the union of the sets is always at most $n$, the average (over all elements), of the number of sets an element belongs to at step $i$ is at least $\frac{Y_i \cdot X}{n}$. Therefore, the element $x_i$ selected at step $i$ must belong to at least this number of sets. Thus, for every $i \geq 1$,

$$Y_i \leq Y_{i-1} - \frac{Y_{i-1} \cdot X}{n} = Y_{i-1} \cdot \left(1 - \frac{X}{n}\right) \leq Y_0 \cdot \left(1 - \frac{X}{n}\right)^i < Y_0 \cdot e^{-i \cdot \frac{X}{n}}$$

By substituting $Y_0$ with $n$ and $X$ with $\frac{2}{\delta}\ln(2/\delta)$, we get that $Y_i < \frac{\delta}{2}n$ for $i = \frac{\delta}{2}n$, and we are done. □.

**Proof of Lemma 4.4** Let $B$ be the set of vertices whose $C$-neighborhood contains less than $k$ vertices. By the Theorem's premise, $|B| \leq \frac{\delta}{2}n$. By definition, for every $v$ and $u$, if $u \in \Gamma_C(v)$ then $v \in \Gamma_C(u)$. Suppose there exists a set $T$ of vertices of size at most $\frac{\delta}{2}n$, such that for every $v \in V \setminus B$, $T \cap \Gamma_C(v) \neq \emptyset$. Namely, the set $T$ is a set of centers of balls of radius $C$ whose union covers all vertices in $V \setminus B$. Then, we can arbitrarily select one vertex in $T$, denoted $v_0$, and add an edge between every other vertex in $T$ and $v_0$ and between every vertex in $B$ and $v_0$. In this way we obtain a graph of diameter at most $2C + 2$ by adding $|T - 1| + |B| \leq \delta n$ edges.

The existence of a set $T$ as desired follows directly from Claim 4.5 and the premise of the Theorem concerning the size of the $C$-neighborhoods of all vertices in $V \setminus B$. Simply let $S_1, \ldots, S_Y$ be the $C$-neighborhood sets of vertices in $V \setminus B$. □

# 5 Proof of Main Theorem

The proofs of Items 1 and 2 in Theorem 3.3 will be shown to follow from the results in the previous section. The proof of the third item is slightly more involved and will require to establish additional claims concerning properties of graphs that have diameter at most $D$. As stated before, all algorithms referred to by the theorem are based on the basic algorithm Test-Diameter. Recall that the algorithm selects $S$ vertices, and accepts if the $C$-neighborhood of at least $(1 - \alpha)S$ of the selected vertices contains $k$ vertices.

**Definition 5.1** *A vertex whose $C$-neighborhood contains less than $k$ vertices will be called* bad.

The basic proof line will be to show that if the graph has diameter at most $D$, then the fraction of bad vertices is small (0 in some cases), and therefore the algorithm will accept (with probability at least 2/3). If the graph is $\epsilon$-far from diameter $\beta(D)$ for the various boundary functions $\beta(\cdot)$, then the sample selected by the algorithm will (with probability at least 2/3), include a fraction greater than $\alpha$ of bad vertices and hence the graph will be rejected.

## 5.1 Proof of Item 1 in Theorem 3.3

Item 1 corresponds to the variant of the algorithm Test-Diameter where $C = D$, $k = \frac{2}{\epsilon_{n,m}}$, and $\alpha = 0$. In this case the sample size will be $S = \frac{4}{\epsilon_{n,m}}$. Namely, (for the non-trivial case of $\frac{1}{\epsilon_{n,m}} = \Omega(\frac{1}{n})$),

the algorithm requires that the $D$-neighborhood of *every* starting vertex selected be of size at least $k = \frac{2}{\epsilon_{n,m}}$. Clearly, every graph with diameter at most $D$ will pass the test, and so it remains to show that graphs that are $\epsilon$-far from diameter $\beta(D) = 4D + 2$ will be rejected with probability at least 2/3.

By Lemma 4.3 if the $D$-neighborhood of at most $\frac{1}{k}n = \frac{\epsilon_{n,m}}{2}n$ of the vertices contains less than $k = \frac{2}{\epsilon_{n,m}}$ vertices, then the graph can be transformed into a graph with diameter at most $4D + 2$ by adding at most $\frac{1}{k} \cdot n = \epsilon_{n,m} \cdot n = \epsilon \cdot m$ edges. This implies that every graph which is $\epsilon$-far from diameter $4D + 2$ must have more than $\frac{\epsilon_{n,m}}{2} \cdot n$ bad vertices. For any such graph, the probability that a sample of size at least $S = \frac{4}{\epsilon_{n,m}}$ will not contain such an "incriminating" bad vertex is at most

$$\left(1 - \frac{\epsilon_{n,m}}{2}\right)^S < e^{-2} < 1/3$$

and and the correctness of the algorithm follows.

## 5.2 Proof of Item 2 in Theorem 3.3

Item 2 corresponds to the variant of the algorithm Test-Diameter where $C = D$, $k = \frac{4}{\epsilon_{n,m}} \cdot \ln \frac{4}{\epsilon_{n,m}}$, $\alpha = 0$, and the sample size is $S = \frac{4}{\epsilon_{n,m}}$. Again, every graph with diameter $D$ is always accepted. We next show that every graph that is $\epsilon$-far from diameter $\beta(D) = 2D + 2$ is rejected with probability at least 2/3. By Lemma 4.4 if the $D$-neighborhood of at least $(1 - \frac{\epsilon_{n,m}}{2})n$ of the vertices contains at least $\frac{4}{\epsilon_{n,m}} \ln \frac{4}{\epsilon_{n,m}}$ vertices, then the graph can be transformed into a graph with diameter at most $2D + 2$ by adding at most $\epsilon_{n,m} \cdot n = \epsilon \cdot m$ edges. Therefore, every graph which is $\epsilon$-far from diameter $2D + 2$ must have more than $\frac{\epsilon_{n,m}}{2} \cdot n$ bad vertices. The proof proceeds as in the previous item.

## 5.3 Proof of Item 3 in Theorem 3.3

For this item we set $C = \frac{D}{2} + \frac{D}{2^{i+1}-2}$, $k = \frac{4}{\epsilon_{n,m}} \ln \frac{4}{\epsilon_{n,m}}$, $\alpha = \frac{\epsilon_{n,m}}{4}$, and $S = \frac{48}{\epsilon_{n,m}}$. Note that $C < D$, and this will be the source of our 2-sided error. Let $\chi_i$ be a random variable, such that $\chi_i = 1$ if the $i$'th vertex sampled is bad, and $\chi_i = 0$ otherwise.

We first show that every graph that is $\epsilon$-far from diameter $\beta(D) = (1 + \frac{1}{2^i-1}) \cdot D + 2$ is rejected with probability at least 2/3. By Lemma 4.4, every graph that is $\epsilon$-far from having diameter at most $2C + 2 = D(1 + \frac{1}{2^i-1}) + 2$ has at least $\frac{\epsilon_{n,m}}{2} \cdot n$ bad vertices. Thus,

$$\Pr[\chi_i = 1] \geq \frac{\epsilon_{n,m}}{2}.$$

By a multiplicative Chernoff bound, if we uniformly select $S \geq \frac{48}{\epsilon_{n,m}}$ vertices, then the probability that we obtain less than a fraction of $\alpha = \frac{\epsilon_{n,m}}{4}$ bad vertices (i.e., less than half the expected fraction), is bounded by

$$\Pr\left[\frac{1}{S}\sum_{i=1}^{S}\chi_i < \alpha\right] = \Pr\left[\frac{1}{S}\sum_{i=1}^{S}\chi_i < \frac{\epsilon_{n,m}}{2}(1 - \frac{1}{2})\right] \leq \exp\left(-\frac{1}{4} \cdot \frac{\epsilon_{n,m}}{2} \cdot S \cdot \frac{1}{2}\right) = e^{-3} < 1/3 \ .$$

Therefore, the graph is rejected with probability at least 2/3.

We now turn to the case in which the graph has diameter at most $D$. The algorithm walks to distance smaller than $D$ from each selected vertex and rejects the graph in case the fraction of selected vertices that reach less than $k$ vertices is greater than $\alpha = \frac{\epsilon_{n,m}}{4}$. Thus, we cannot claim anymore that a graph with diameter $D$ will always be accepted. Instead, we shall want to bound the number (fraction) of vertices whose $C$-neighborhoods are small. We show:

**Lemma 5.2** *For every graph with diameter at most $D$, for every $C$ of the form $C = \frac{D}{2} + \frac{D}{2^{i+1}-2}$ where $i \geq 2$, and for every $k$, the number of vertices whose $C$-neighborhood is of size less than $k$ is at most $k^{i+1}$.*

We prove the lemma momentarily, but first show how the completeness of Item 3 in Theorem 3.3 follows (i.e., that every graph with diameter at most $D$ is accepted with probability at least 2/3). By the lemma, the number of bad vertices of every graph with diameter at most $D$ is at most $k^{i+1} = \left(\frac{4}{\epsilon_{n,m}} \cdot \log \frac{4}{\epsilon_{n,m}}\right)^{i+1}$. By our assumption on $\epsilon$ as a function of $n$ and $m$, we have that $\epsilon_{n,m} = \Omega(n^{-\frac{1}{i+2}} \cdot \log n/(i+2))$. Therefore, for the appropriate setting of the constants in the $\Omega(\cdot)$ notation, we get that the number of bad vertices is at most

$$\left(\frac{4}{\epsilon_{n,m}} \cdot \log \frac{4}{\epsilon_{n,m}}\right)^{i+1} < \frac{\epsilon_{n,m}}{8} \cdot n.$$

Thus, in this case:

$$\Pr[\chi_i = 1] < \frac{\epsilon_{n,m}}{8}.$$

Therefore, by a multiplicative Chernoff bound, the probability that the algorithm rejects is at most:

$$\Pr\left[\frac{1}{S}\sum_{i=1}^{S}\chi_i > \alpha\right] = \Pr\left[\frac{1}{S}\sum_{i=1}^{S}\chi_i > \frac{\epsilon_{n,m}}{8}(1+1)\right] \leq \exp\left(-\frac{1}{3}\frac{\epsilon_{n,m}}{8} \cdot S\cdot\right) = e^{-2} < 1/3$$

and this completes the proof of Item 3.

**Proof of Lemma 5.2:** Assume, contrary to the claim that there are more than $k^{i+1}$ bad vertices whose $C$-neighborhood contains less than $k$ vertices. Since $C > \frac{D}{2}$, it follows that the $\frac{D}{2}$-neighborhood of all bad vertices contains less than $k$ vertices as well. Since the graph has diameter at most $D$, every two vertices must have at least one common vertex in their respective $\frac{D}{2}$-neighborhoods. Let us fix some bad vertex $u$. By the above, $u$ has less than $k$ vertices in its $\frac{D}{2}$-neighborhood, and it shares at least one vertex with every one of the other $k^{i+1}$ bad vertices. Therefore, there exists at least one vertex $v$ in $u$'s $\frac{D}{2}$-neighborhood that belongs to the $\frac{D}{2}$-neighborhood of at least $k^i$ additional bad vertices.

Let these vertices be denoted $u_1, \ldots, u_t$, and consider a tree $T_v$ of height at most $\frac{D}{2}$ rooted at $v$ and containing all these vertices. Namely, we construct the tree in at most $t$ steps. In each step we add a new vertex $u_j$ (among $u_1, \ldots, u_t$) that does not yet belong to the tree, and a path of length at most $\frac{D}{2}$ from $u_j$ to $v$. (If the new path forms a cycle we may always remove the new edges that create the cycle with the edges already in the tree.) The resulting tree has height $h \leq \frac{D}{2}$, contains at least $t \geq k^i$ vertices, and all its leaves belong to $\{u_1, \ldots, u_t\}$. We make the following claim.

**Claim 5.3** *For any tree of height $h$ and size at least $t$, and for any $a < h$, there exists a leaf in the tree whose $(h+a)$-neighborhood contains at least $t^{\frac{1}{\log((h+a)/a)+1}}$ vertices.*

We prove the claim momentarily, but first show how it can be applied to complete the proof of Lemma 5.2. Let $a = \frac{D}{2^{i+1}-2}$, so that $C = \frac{D}{2} + a$. Clearly, the $C$-neighborhood (in $G$) of every bad vertex $u_j$, contains at least as many vertices as the $C$-neighborhood of $u_j$ restricted to the tree $T_v$. But according to Claim 5.3, there exists at least one bad vertex $u_j$, that is a leaf in $T_v$, whose $C$-neighborhood in the tree contains at least

$$t^{\frac{1}{\log(2^i)+1}} = t^{\frac{1}{i+1}} \geq k$$

contradicting the fact that $u_j$ is bad. Lemma 5.2 follows.

**Proof of Claim 5.3:** Before we prove the claim in general, we consider a special case that will serve as an introduction to the general case. Let $a = \frac{h}{3}$. We show that there exists at least one leaf whose $\frac{4}{3}h$-neighborhood contains at least $\sqrt{t}$ vertices. We consider two cases. In case the $\frac{1}{3}h$-neighborhood of the root contains at least $\sqrt{t}$ vertices, we are done, since each leaf can reach each vertex in the $\frac{1}{3}h$-neighborhood of the root in at most $h + \frac{1}{3}h = \frac{4}{3}h$ steps. Otherwise, the number, $b$, of vertices that are at distance at most $\frac{1}{3}h$ from the root is less than $\sqrt{t}$. Consider the sub-trees rooted at the vertices that are at distance (exactly) $\frac{1}{3}h$ from the root. Since the number of such sub-trees is bounded by $b-1$ there must be at least one sub-tree with at least

$$\frac{t-b}{b-1} + 1 = \frac{t-1}{b-1} \geq \frac{t-1}{\sqrt{t}-2} > \sqrt{t}$$

vertices (the $+1$ is due to the root of the sub-tree). But the height of each such sub-tree is at most $h - \frac{1}{3}h = \frac{2}{3}h$, and so each leaf in the sub-tree can reach every other node in the sub-tree in at most $2 \cdot \frac{2}{3}h = \frac{4}{3}h$ steps, and the claim for $a = \frac{h}{3}$ follows.

For the general case we define a function $f(t,h,a)$ which is the minimum over all trees $T$ of height $h$ having $t$ vertices, of the maximum size of the $(h+a)$-neighborhood of a leaf in the tree (where the neighborhood is restricted to the tree). That is:

$$f(t,h,a) = \min_{T} \max_{u \ a \ leaf} |\Gamma_{h+a}(u)|.$$

Let $s(h,a) \stackrel{\text{def}}{=} \lceil \log((h+a)/a) \rceil$. We claim that for every $a \geq 1$, $f(t,h,a) \geq t^{\frac{1}{s(h,a)}}$, and prove the claim by induction.

First note that we may assume that $t > h$ (as every tree having height $h$ has at least $h+1$ vertices). If either $t = 1$ or $a \geq h$, then the claim clearly holds, as in these cases $f(t,h,a) = t$. Thus consider any $h \geq 1$, $t > h$ and $a < h$ and assume the claim holds for every $t' < t$, $h' < h$, and $a' > a$. Let us look at any leaf $w$ in the tree. The $(h+a)$-neighborhood of $w$ contains all vertices at distance at most $a$ from the root. Furthermore, if the number of these vertices is $b$, then there exists at least one sub-tree of height $h-a$, rooted at one of the vertices at distance $a$ from the root that contains at least $\frac{t-b}{b-1} + 1 = \frac{t-1}{b-1}$ vertices. To see why this is true, observe that there are $t-b$ vertices at distance greater than $a$ from the root, and they belong to at most $(b-1)$ sub-trees. Thus, there exists at least one sub-tree with at least $\frac{t-b}{b-1}$ vertices at distance greater than $a$ from the root. Adding the root of this sub-tree (at distance exactly $a$ from the root), we get the above expression. Thus, there exists at least one leaf that can reach $f(\frac{t-1}{b-1}, h-a, 2a)$ vertices in its sub-tree. Taking care not to count the root of this sub-tree twice, we get

$$f(t,h,a) \geq \min_{b>1} \left( b - 1 + f\left( \frac{t-1}{b-1}, h-a, 2a \right) \right) . \tag{2}$$

By the induction hypothesis we have

$$f(t,h,a) \geq \min_{b>1} \left( b - 1 + \left( \frac{t-1}{b-1} \right)^{\frac{1}{s(h-a,2a)}} \right) \tag{3}$$

We next show that $f(t,h,a) \geq t^{\frac{1}{s+1}}$, where $s$ is used here as a shorthand for $s(h-a,2a)$. By taking a derivative of the function in the right hand side of Equation (3), we find that for any fixed $t > 1$,

$s \geq 1$, the minimum of $\left(b - 1 + \left(\frac{t-1}{b-1}\right)^{\frac{1}{s}}\right)$ is obtained at $b - 1 = (t-1)^{\frac{1}{s+1}} s^{-\frac{s}{s+1}}$, and the value at this point is

$$(t-1)^{\frac{1}{s+1}} \cdot \left(s^{\frac{1}{s+1}} + s^{-\frac{s}{s+1}}\right) . \tag{4}$$

Recall that $t$ and $s$ are both integers, where $t > 1$ and $s \geq 1$. For $s = 1$ the expression in Equation (4) is $(t-1)^{\frac{1}{2}} \cdot 2$ which is at least $t^{\frac{1}{2}} = t^{\frac{1}{s+1}}$ for every integer $t > 1$. For any $s \geq 2$, the expression in Equation (4) is at least $(s(t-1))^{\frac{1}{s+1}}$ which again is at least $t^{\frac{1}{s+1}}$ for every integer $t > 1$. By Equation (3), the bound on $f(t, h, a)$ follows

It remains to show that $t^{\frac{1}{s(h-a,2a)+1}} \geq t^{\frac{1}{s(h,a)}}$. If $\frac{h+a}{2a}$ is an exponent of 2, that is $\frac{h+a}{2a} = 2^i$ for some $i \geq 0$, then

$$s(h-a, 2a) + 1 = \left\lceil \log\left(\frac{h+a}{2a}\right)\right\rceil + 1 = i + 1 = \left\lceil \log\left(\frac{h+a}{a}\right)\right\rceil = s(h, a)$$

and the induction step is completed. Otherwise, $\frac{h+a}{2a} = 2^{i-1} + X$, where $i \geq 1$, and $0 < X < 2^{i-1}$ (recall that $h > a$ so $h + a > 2a$). In this case it is still true that $\left\lceil \log\left(\frac{h+a}{2a}\right)\right\rceil + 1 = i + 1$, and $\left\lceil \log\left(\frac{h+a}{a}\right)\right\rceil = \lceil \log(2^i + 2X)\rceil = i + 1$, and the Claim follows. □

**Remark:** We note that the above Claim is almost tight for our application of $a = 1$. Namely, there exist trees of height $h$ and size $t$, such that the $(h+1)$-neighborhood of every leaf contains $O(t^{\frac{1}{\log h - 1}} \cdot h)$ vertices. Such trees have the following form. They are constructed of $\log(h/2)$ *super-layers*, where the $i$'th super-layer (counting from the root of the tree) consists of $2^i$ layers. The vertices in the first layer of each super-layer have $b \approx (t/h)^{\frac{1}{\log(h/2)}}$ children, and all other vertices have a single child. Hence, the tree is a kind of "star with splits". By this construction, the $h+1$ neighborhood of every leaf contains roughly

$$b \cdot \frac{h}{2} + b \cdot \frac{h}{4} + \ldots + b \;\leq\; b \cdot h \;\approx\; t^{\frac{1}{\log h - 1}} \cdot h^{1 - \frac{1}{\log h}} .$$

For an illustration, see Figure 2.

# 6 Transforming Results from the Bounded-Degree Model

As mentioned in the introduction, many of the results for the bounded-degree model can be transformed to our unbounded-degree model. In this section we sketch the transformations that exist, and point out the difficulties for the results that cannot be transformed.

## 6.1 Testing $k$-Connectivity and Testing whether a Graph is Eulerian

We first present the algorithm for testing connectivity (that is, $k = 1$). Let $t$ and $s$ be functions of $\epsilon$ (and possibly $n$ and $m$) that will be determined momentarily. Let $\epsilon_{n,m} = \frac{m}{n} \cdot \epsilon$ be as defined previously. As in [GR97, Alg. 3.4], here too the algorithm uniformly and independently selects $t$ starting vertices. From each starting vertex, the algorithm performs a Breadth-First-Search (BFS), until it sees $s$ vertices, or it cannot reach any more vertices. In the latter case, a small connected component is discovered (implying that the graph is not connected), and the algorithm rejects the graph. Otherwise, no small connected component is discovered, and the algorithm accepts.

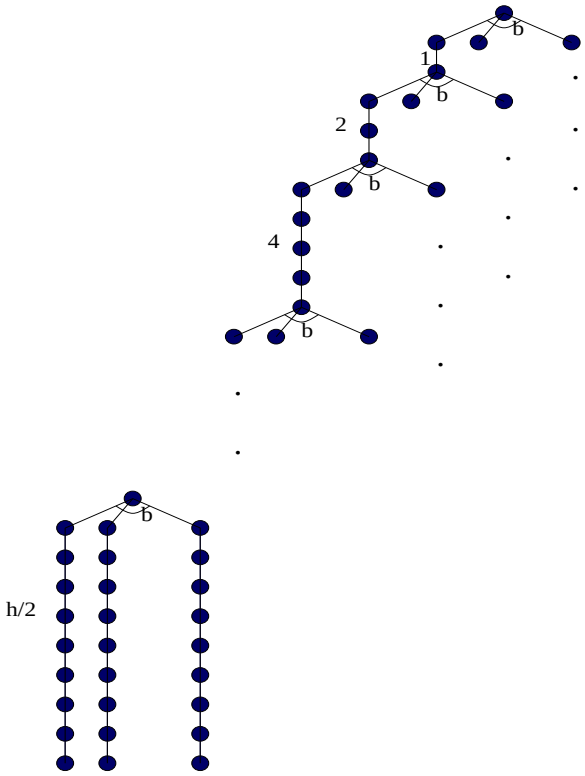


Figure 3: A tree that approximately matches the bound in Theorem 5.3 for $a = 1$ (only the left-most splits in the tree are drawn).

The only differences between the algorithm in the bounded-degree model and the algorithm in our model, is the setting of $t$ and $s$, and the bound on the running time of the algorithm. The key observation is that any graph which is $\epsilon$-far from being connected has "many" connected components. In our case, "many" is more than $\epsilon m$ (as otherwise, by adding at most $\epsilon m - 1$ edges, it is possible to make the graph connected). This implies that if the graph $\epsilon$-far from being connected, then there exist at least $\frac{\epsilon}{2}m$ components with less than $\frac{2}{\epsilon_{n,m}}$ vertices each. By setting $t = \frac{4}{\epsilon_{n,m}}$ and $s = \frac{2}{\epsilon_{n,m}}$, we obtain the desired success probability. Since each BFS requires at most $s^2$ queries, the resulting query complexity and running time of the algorithm are $O(1/\epsilon_{n,m}^3)$. An improved version of the algorithm (see [GR97, Alg. 3.5]), has query complexity and running time $O\left(\frac{\log(1/\epsilon_{n,m}))}{(\epsilon_{n,m})^2}\right)$ (compared to $O\left(\frac{\log^2(1/(\epsilon d))}{\epsilon}\right)$ in the bounded-degree model, where $d$ is a bound on the degree).

The algorithms for $k = 2$ and $k = 3$, are modified similarly to the algorithm described above for $k = 1$. For $k = 2$ we get the same complexities as for $k = 1$. For $k = 3$ the bound increases by a factor of $1/\epsilon_{n,m}$.

For a general $k$, the heart of the algorithm in [GR97] is a random search process (that is repeated several times from each of the $t$ uniformly selected starting vertices). The goal of the random process is to detect whether the starting vertex belongs to a small set that is connected to the rest of the graph by less than $k$ edges (in this case the graph is not $k$-connected). Let $S$ be the set of vertices visited so far by the algorithm. In each step the algorithm selects at random (as specified below) an edge among the cut edges in $(S, \overline{S})$, and adds to $S$ the new vertex reached by this edge. This process is repeated as long as $|S| \leq s$ and the size of the cut $(S, \overline{S})$ is at least $k$.

If at some point the size of the cut $(S, \overline{S})$ is less than $k$, then the graph is rejected. The random selection of edges is done as follows. When a new vertex $v$ is added to $S$, the edges incident to $v$ that were not yet assigned costs, are each assigned a random cost uniformly in $[0, 1]$. At each step the algorithm selects the edge with the lowest cost in $(S, \overline{S})$.

The analysis of this procedure remains the same in our model. Specifically, the number of independent executions of the procedure (from each starting vertex) that ensure a constant probability of detecting a small cut (if one exists), is $O(s^2)$. With the appropriate choice of $t$ and $s$, the correctness of the algorithm can be shown to follow.[2] It remains to deal with the implementation of the random search procedure. Since we do not have a degree-bound, it may seem that assigning costs to all edges incident to a vertex may be too expensive. However, we need only note that if a vertex has degree greater than $s + k$, then it clearly cannot belong to a set of size at most $s$ that is connected to the rest of the graph by less than $k$ edges. Hence, in case such a vertex is encountered, we may stop the process (without rejection). The complexity of the process is therefore $O(s^2)$. By setting $t = O(k/\epsilon_{n,m})$, and $s = O(k/\epsilon_{n,m})$ (and recalling that the process is repeated $O(s^2)$ times from each starting vertex), the complexity of (the basic version of) the algorithm is $O(k^5/\epsilon_{n,m}^5)$. A slight modification of the algorithm (as done in [GR97]), has complexity $\tilde{O}(k^4/\epsilon_{n,m}^4)$.

**Testing whether a Graph is Eulerian.** The algorithm in [GR97] first invokes the connectivity testing algorithm with distance parameter $\epsilon/2$. If the connectivity test fails then the graph is rejected by the algorithm. Otherwise, the algorithm uniformly selects $t = O(1/(\epsilon d))$ vertices and rejects if more than two vertices are found to have odd degree. The algorithm can be easily modified by simply setting $t = O(1/\epsilon_{n,m})$. The analysis remains almost exactly the same.

### 6.2 Testing Planarity and Testing whether a Graph is Cycle-Free.

The planarity-testing algorithm of [GR97] is based on an algorithm for testing whether a graph does not contain a particular sub-graph (here, $K_{3,3}$ or $K_5$). The sub-graph freeness testing algorithm performs a BFS up to depth that is the diameter of the sub-graph. Hence its complexity has an explicit (polynomial) dependence on the degree bound $d$, and it is not applicable in our unbounded-degree case.

In contrast, the complexity of Cycle-Freeness algorithm does not have an explicit dependence on the degree-bound (more precisely, it has complexity $O(1/(\epsilon^3 d)) < O(1/\epsilon^3)$). However, since the algorithm approximates the number of edges in a sub-graph of the tested graph, it has an implicit dependency on such a bound. Furthermore, *any* algorithm for Cycle-Freeness in the general model requires $\Omega(\sqrt{n})$ queries. This follows from the fact that in fewer queries an algorithm cannot distinguish between the following two graphs (or more precisely, families of graphs, as the graphs are defined up to isomorphism). The first graph is the empty graph, which is clearly cycle free. The second graph consists of a clique of size $\sqrt{n}$ and an independent set on the remaining $n - \sqrt{n} vertices$. This graph is $\Theta(1)$-far from being cycle free as it has $\Theta(n)$ edges, and we need to remove $\Theta(n)$ edges from within the clique to make it cycle free.

### 6.3 Hardness Results

All hardness results (both for query complexity and for time complexity) clearly transfer to our model as it is more general. In fact, the proof of the lower bound of $\Omega(\sqrt{n})$ for testing bipartitness is much easier in our model. This holds simply since in fewer queries an algorithm cannot distinguish

[2] In fact, in our model the proof of correctness is simplified, since we are not restricted by a degree-bound.

between the empty graph (which is clearly bipartite), and a graph consisting of a clique of size $\sqrt{n}$ and an independent set on the remaining $n - \sqrt{n}$ vertices (which is $\Theta(1)$-far from being bipartite).

### Acknowledgments

We would like to thank Noga Alon for bringing [AGR99] to our attention. We would also like to thank two anonymous referees for their helpful comments.

## References

[AGR99] N. Alon, A. Gyárfás, and M. Ruszinkó. Decreasing the diameter of bounded degree graphs. To appear in Journal of Graph Theory, 1999.

[AP90] B. Awerbuch and D Peleg. Sparse partitions. In *Proceedings of the Thirty-First Annual Symposium on Foundations of Computer Science*, pages 503–513, 1990.

[AS92] N. Alon and J. H. Spencer. *The Probabilistic Method*. John Wiley & Sons, Inc., 1992.

[Awe85] B. Awerbuch. Complexity of network synchronization. *Journal of the Association for Computing Machinery*, 32:804–823, 1985.

[DF85] M. E. Dyer and A. M. Frieze. A simple heuristic for the $p$-centre problem. *Operations Research Letters*, 3(6):285–288, 1985.

[GGR98] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the Association for Computing Machinery*, 45(4):653–750, 1998. An extended abstract appeared in FOCS96.

[GR97] O. Goldreich and D. Ron. Property testing in bounded degree graphs. In *Proceedings of the Thirty-First Annual ACM Symposium on the Theory of Computing*, pages 406–415, 1997.

[KR98] M. Kearns and D. Ron. Testing problems with sub-learning sample complexity. In *Proceedings of the Eleventh Annual ACM Conference on Computational Learning Theory*, pages 268–279, 1998.

[LMSL92] C. L. Li, S. T. McCormick, and D. Simchi-Levi. On the minimum cardinality bounded diameter and the bounded cardinality minimum diameter edge addition problems. *Operations Research Letters*, 11(5):303–308, 1992.

[LS93] N. Linial and M. Saks. Low diameter graph decompositions. *Combinatorica*, 13:441–454, 1993.

[RS96] R. Rubinfeld and M. Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.