AN ALGORITHM TO EVALUATE THE SPECTRAL EXPANSION

HAU-WEN HUANG

Abstract. Assume that \( X \) is a connected \((q + 1)\)-regular undirected graph of finite order \( n \). Let \( A \) denote the adjacency matrix of \( X \). Let \( \lambda_1 = q + 1 > \lambda_2 \geq \lambda_3 \geq \ldots \geq \lambda_n \) denote the eigenvalues of \( A \). The spectral expansion of \( X \) is defined by

\[
\Delta(X) = \lambda_1 - \max_{2 \leq i \leq n} |\lambda_i|.
\]

By the Alon–Boppana theorem, when \( n \) is sufficiently large, \( \Delta(X) \) is quite high if \( \mu(X) = q - \frac{1}{2} \max_{2 \leq i \leq n} |\lambda_i| \) is close to 2. In this paper, with the inputs \( A \) and a real number \( \varepsilon > 0 \) we design an algorithm to estimate if \( \mu(X) \leq 2 + \varepsilon \) in \( O(n^{\omega \log \log 1+\varepsilon} n) \) time, where \( \omega < 2.3729 \) is the exponent of matrix multiplication.

Keywords: dynamic programming, geodesic cycle, Ihara zeta function, spectral expansion.

MSC2020: 05C30, 05C48, 05C50.

1. Introduction

Throughout this paper we adopt the following conventions: The function \( \log \) without a base refers to base 2. The notation \( \omega \) stands for the exponent of matrix multiplication. The improvements of the Coppersmith–Winograd algorithm \( [4,9,22] \) indicate that \( \omega < 2.3729 \). Let \( q \geq 1 \) and \( n \geq 2 \) be two integers. We assume that \( X \) is a connected \((q + 1)\)-regular undirected graph with \( n \) vertices. Let \( A \) denote the adjacency matrix of \( X \). Let

\[
\lambda_1 = q + 1 > \lambda_2 \geq \lambda_3 \geq \ldots \geq \lambda_n
\]

denote the eigenvalues of \( A \).

Roughly speaking, the graph \( X \) is a \textit{good expander} if the parameter \( q \) is low and the expansion parameters are high. Such graphs are widely studied in mathematics and have several applications to computer science such as networks, error-correcting codes and probabilistic algorithms \( [1,5,11,16,21] \). There are two common expansion parameters called the spectral expansion and the edge expansion. The \textit{spectral expansion} of \( X \) is measured by the spectral gap of \( X \):

\[
\Delta(X) = \lambda_1 - \max_{2 \leq i \leq n} |\lambda_i|.
\]

Given any two sets \( S, T \) of vertices of \( X \), let \( E(S, T) \) denote the set of all edges with one vertex in \( S \) and the other vertex in \( T \). The \textit{edge expansion} of \( X \) is given by

\[
h(X) = \min_{1 \leq |S| \leq n/2} \frac{|E(S, \overline{S})|}{|S|}
\]

where \( \min \) is over all sets \( S \) of vertices of \( X \) with \( 1 \leq |S| \leq \frac{n}{2} \). The expander mixing lemma \( [2] \) implies that \( |E(S, T)| \) will be closer to the expected number \( \frac{(q+1)|S||T|}{n} \) for all sets \( S, T \) of vertices of \( X \), provided that \( \Delta(X) \) is as large as possible. In \( [7] \) an inequality states that

\[
\frac{q - \lambda_2 + 1}{2} \leq h(X) \leq \sqrt{2(q + 1)(q - \lambda_2 + 1)}.
\]
We set
\[ \mu(X) = q^{-\frac{1}{2}} \max_{2 \leq i \leq n} |\lambda_i|. \]

The Alon–Boppana theorem \cite{8,18} implies that
\[ \mu(X) \geq q^{-\frac{1}{2}} \lambda_2 \geq 2 - o(1) \]

where the term \( o(1) \) is a quantity that tends to 0 for every fixed \( q \) as \( n \) approaches \( \infty \). Thus, when \( n \) is sufficiently large, the expansion parameters \( \Delta(X) \) and \( h(X) \) are quite high provided that \( \mu(X) \) is close to 2.

To obtain \( \mu(X) \), one may compute \( \lambda_i \) for all \( i = 2, 3, \ldots, n \) via the QR algorithm or Jacobi eigenvalue algorithm. Each iteration step of both algorithms has \( O(n^3) \) time complexity \cite{18}. In this paper, we introduce a certain number sequence \( \{H_k\}_{k=1}^{\infty} \) and investigate its connection with \( \mu(X) \). Furthermore, we design an algorithm to estimate if \( \mu(X) \) is close to 2 via the sequence \( \{H_k\}_{k=1}^{\infty} \) in \( o(n^3) \) time. To define \( \{H_k\}_{k=1}^{\infty} \) we begin by recalling the notation of geodesic cycles. We endow two opposite orientations on all edges of \( X \) called the oriented edges of \( X \). Recall that a walk
\[ P = (e_1, e_2, \ldots, e_k) \]
is a sequence of oriented edges of \( X \) such that the end vertex of \( e_i \) is the start vertex of \( e_{i+1} \) for all \( i = 1, 2, \ldots, k - 1 \). A walk \( P \) is called a cycle if the start vertex of \( e_1 \) is the end vertex of \( e_k \). A walk \( P \) is said to have backtracking if \( e_{i+1} \) is the oriented edge opposite to \( e_i \) for some \( i = 1, 2, \ldots, k - 1 \). A walk is backtrackless if it has no backtracking. A cycle is geodesic if its all shifted cycles are backtrackless \cite{13,15}. Let \( N_k \) denote the number of geodesic cycles on \( X \) of length \( k \geq 1 \). The numbers \( \{N_k\}_{k=1}^{\infty} \) first appeared in the Ihara zeta function \( Z(u) \) of \( X \) which is defined as the analytic continuation of
\[ \exp \left( \sum_{k=1}^{\infty} \frac{N_k}{k} u^k \right). \]

The Ihara zeta function \( Z(u) \) of \( X \) was first considered by Y. Ihara \cite{14} in the context of discrete groups. As suggested by J.-P. Serre, \( Z(u) \) has a graph-theoretical interpretation \cite{19}. The graph \( X \) is called Ramanujan \cite{17} whenever
\[ \max_{|\lambda_i| < q+1} |\lambda_i| \leq 2q^{\frac{1}{2}} \]

where \( \max \) is over all \( i = 1, 2, \ldots, n \) with \( |\lambda_i| < q + 1 \). It was discovered by T. Sunada \cite{20} that \( Z(u) \) satisfies an analogue of the Riemann hypothesis if and only if \( X \) is a Ramanujan graph. A recent result \cite{12} provides a necessary and sufficient condition for \( X \) as Ramanujan in terms of Hasse–Weil bounds on \( \{N_k\}_{k=1}^{\infty} \). Inspired by \cite{12} we consider the numbers
\[ H_k = \begin{cases} 2(n - 1) + q^{\frac{k}{2}} + q^{-\frac{k}{2}} - q^{-\frac{k}{2}} N_k & \text{if } k \text{ is odd}, \\ 2(n - 1) + q^{\frac{k}{2}} + q^{-\frac{k}{2}} - q^{-\frac{k}{2}} (N_k - n(q - 1)) & \text{if } k \text{ is even} \end{cases} \]

for all integers \( k \geq 1 \).

The paper will proceed as follows. In \S 2 we design a dynamic programming algorithm that computes \( N_k \) in \( O(n^\omega \log k) \) time. Moreover we modify the algorithm to compute \( H_k \) with the same time complexity. In \S 3 we relate \( \{H_k\}_{k=1}^{\infty} \) to \( \mu(X) \). In \S 4 with the inputs \( A \) and a real number \( \varepsilon > 0 \) we describe our main algorithm which evaluates if \( \mu(X) \leq 2 + \varepsilon \) in \( O(n^\omega \log \log_{1+\varepsilon} n) \) time.
2. The Computations of \( N_k \) and \( H_k \)

Recall that the directed edge matrix \( W \) of \( X \) is the \((0,1)\)-matrix indexed by the oriented edges of \( X \) such that \( W_{ef} = 1 \) if and only if the end vertex of \( e \) is the start vertex of \( f \) and \( e \) is not the oriented edge opposite to \( f \) for all oriented edges \( e, f \) of \( X \). Observe that

\[
N_k = \text{trace}(W^k) \quad \text{for all integers } k \geq 1.
\]

Note that (2) holds for irregular graphs. Let \( m = n(q + 1) \) denote the number of oriented edges of \( X \). In other words, this establishes the following formula [3, 10, 21]:

\[
Z(u)^{-1} = \det(I_m - Wu).
\]

To obtain \( W^k \), it only uses at most \( 2^\lceil \log k \rceil \) times of \( m \times m \) matrix multiplication by applying binary exponentiation. Thus, along the vein (2) the computation of \( N_k \) takes \( O(m^{\omega \log k}) \) time.

For the rest of this paper, let \( \alpha_i \) denote a root of

\[
u^2 - q^{-\frac{1}{2}}\lambda_iu + 1 \quad \text{for all } i = 1, 2, \ldots, n.
\]

In addition to (3) the Ihara zeta function \( Z(u) \) of \( X \) has the following celebrated formula [3, 14]:

\[
Z(u)^{-1} = (1 - u^2)^{\frac{n(q-1)}{2}} \det(I_n - Au + qI_nu^2).
\]

We change the variable \( u \) to \( q^{-\frac{1}{2}}u \) and then take logarithm and differential on either side of (4). Multiplying the resulting equation by \( u \) yields that

\[
\sum_{k=1}^{\infty} q^{-\frac{1}{2}}N_ku^k \text{ is equal to } \frac{n(q-1)}{2} \sum_{k=1}^{\infty} (q^{-\frac{1}{2}} + (-q)^{-\frac{1}{2}})u^k \text{ plus }
\]

\[
\sum_{i=1}^{n} \sum_{k=1}^{\infty} (\alpha_i^k + \alpha_i^{-k})u^k.
\]

Define a family of polynomials \( \{T_k(x)\}_{k=0}^{\infty} \) by

\[
T_{k+1}(x) = xT_k(x) - T_{k-1}(x) \quad \text{for all } k \geq 1
\]

with \( T_0(x) = 2 \) and \( T_1(x) = x \). Note that \( \frac{x}{2}T_k(2x) \) is the \( k \)th Chebyshev polynomial of the first kind for all integers \( k \geq 0 \). Using (6) a routine induction shows that

\[
T_k(x + x^{-1}) = x^k + x^{-k} \quad \text{for all integers } k \geq 0.
\]

It follows from (7) that (5) is equal to

\[
\sum_{i=1}^{n} \sum_{k=1}^{\infty} T_k(q^{-\frac{1}{2}}\lambda_i)u^k = \sum_{k=1}^{\infty} \sum_{i=1}^{n} T_k(q^{-\frac{1}{2}}\lambda_i)u^k = \sum_{k=1}^{\infty} \text{trace}(T_k(q^{-\frac{1}{2}}A))u^k.
\]

Hence we obtain that

\[
N_k = \begin{cases} 
q^{\frac{k}{2}}\text{trace}(T_k(q^{-\frac{1}{2}}A)) & \text{if } k \text{ is odd}, \\
n(q-1) + q^{\frac{k}{2}}\text{trace}(T_k(q^{-\frac{1}{2}}A)) & \text{if } k \text{ is even}.
\end{cases}
\]

Based on the formula (8), we develop a more efficient algorithm to compute \( N_k \). The pseudocode is as follows:
NGC($A, k$)

1. $n = \text{the number of rows of } A$
2. $q = (\text{a row sum of } A) - 1$
3. $L = \text{INDICES}(k)$
4. $l = \text{the length of } L$
5. let $T[0..3]$ denote a new array with $T[0] = A$
6. let $Q[0..3]$ denote a new array with $Q[0] = 1$
7. for $i = l - 1$ downto 1
   8. for $j = 3$ downto 1
      9. $Q[j] = Q[j-1]$
      10. $T[j] = T[j-1]$
      11. if $L[i-1]$ is even
          12. if $L[i-1] = 2L[i]$
              13. $j = 1$
          14. else $j = 2$
          15. else $L[i + j - 1]$ is even
              16. $Q[0] = Q[j]^2$
          17. else $Q[0] = qQ[j]^2$
          18. $T[0] = T[j]^2 - 2Q[0]I_n$
      19. else
          20. if $L[i-1] = 2L[i+1] - 1$
              21. $j = 2$
          22. else $j = 1$
          23. $Q[0] = Q[j]Q[j+1]$
          24. $T[0] = T[j]T[j+1] - Q[0]A$
      25. if $L[0]$ is odd
      26. return $\text{trace}(T[0])$
      27. else return $n(q-1) + \text{trace}(T[0])$

INDICES($k$)

1. let $L$ denote a new empty array
2. $L.\text{append}(k)$  // $L.\text{append}()$ means to add the parameter to the end of $L$
   // the indices of $L$ start with 0
3. while $k$ is even
   4. $k = \frac{k}{2}$
   5. $L.\text{append}(k)$
6. while $k \neq 1$
   7. $k = \frac{k+1}{2}$
   8. $L.\text{append}(k)$
   9. $L.\text{append}(k - 1)$
10. if $k$ is even
    11. $k = k - 1$
12. return $L$

The NGC procedure is a bottom-up dynamic programming algorithm. We are now going to prove the correctness of NGC and analyze its running time.

Lemma 2.1. Let $L$ denote the output array of INDICES($k$). For any entry $L[i] > 1$ the following hold:

(i) If $L[i]$ is even then $L[i + 1]$ or $L[i + 2]$ is equal to $\frac{L[i]}{2}$. 
Lemma 2.1(ii) lines 12–14 make

At the start of each iteration of the loop. Suppose that some of loop sets \( L[i] > 2 \) or line 9 sets \( L[i] \), then \( L[i+1] = \frac{L[i]+1}{2} \) is built in the next iteration. If line 8 sets \( L[i] = 2 \) then line 9 makes \( L[i+1] = 1 \) immediately. Therefore (i) follows.

(ii): Suppose that \( L[i] > 1 \) is even. If each of \( L[0..i] \) is even then \( L[i+1] = \frac{L[i]}{2} \) by the first while loop. Suppose that some of \( L[0..i] \) is odd. Then \( L[i] \) is created by line 8 or 9. If line 8 sets \( L[i] \), then \( L[i+2] = \frac{L[i]+1}{2} \) is built in the next iteration. If line 8 sets \( L[i] = 2 \) then line 9 makes \( L[i+1] = 1 \) immediately. Therefore (i) follows.

Proof. By INDICES procedure the value 1 is only stored in the last entry of \( L \). Since line 8 sets \( Q[0] \) to be 1 initially, it is true prior to the first iteration.

To see each iteration preserves the loop invariant, we suppose that (by the loop invariant. Combining the above comments, lines 12–14 make \( L[i+j-1] = \frac{L[i-1]}{2} \).

\[
Q[j] = q^{[L[i+j-1]/2]}
\]

by the loop invariant. Combining the above comments, lines 15–17 put \( q^{L[i-1]/2} \) in \( Q[0] \).

Now suppose that \( L[i-1] \) is odd. Lines 8–10 move \( Q[1], Q[0] \) to \( Q[2], Q[1] \) respectively. By Lemma 2.1(ii) lines 20–22 make \( (L[i+j-1], L[i+j]) = (\frac{L[i-1]+1}{2}, \frac{L[i-1]-1}{2}) \). Thus

\[
Q[j], Q[j+1] = \left(q^{L[i+j-1]/2}, q^{L[i+j]/2}\right)
\]

by the loop invariant. Combining the above comments, line 23 places \( q^{L[i-1]/2} \) in \( Q[0] \).

Decrementing \( i \) for the next iteration, the loop invariant is maintained. The lemma follows. \( \square \)

Applying (7) it is routine to verify that

\[
(iii) \quad T_{i+j}(x) = T_i(x)T_j(x) - T_{j-i}(x) \quad \text{for all integers } i, j \text{ with } 0 \leq i \leq j.
\]

Lemma 2.3. At the start of each iteration of the for loop of lines 7–24 of NGC(A,k), the entry

\[
T[0] = q^{L[i]/2} T_{L[i]}(q^{-1/2}A).
\]

Proof. Prior to the first iteration of the loop, \( T[0] = A \) and \( L[i] = 1 \). Hence the loop invariant holds for the first time.

To see each iteration preserves the loop invariant, we suppose that \( L[i-1] \) is even first. By Lemma 2.1(i) lines 12–14 make \( L[i+j-1] = \frac{L[i-1]}{2} \). Combined with the loop invariant, lines 8–10 set \( T[j] = q^{-L[i-1]/4} T_{L[i-1]}(q^{-1/2}A) \). Hence line 18 makes

\[
T[0] = q^{L[i-1]/2} T_{L[i-1]}(q^{-1/2}A)^2 - 2Q[0]I_n = q^{L[i-1]/2} \left(T_{L[i-1]}(q^{-1/2}A)^2 - 2I_n\right) \quad \text{(by Lemma 2.2)}
\]

\[
= q^{L[i-1]/2} T_{L[i-1]}(q^{-1/2}A) \quad \text{(by Equation (9))}.
\]
Now suppose that $L[i-1]$ is odd. By Lemma 2.1(ii), lines 20–22 make $(L[i+j-1], L[i+j]) = (L[i+j-1] + 1, L[i+j-1] - 1)$. Combined with the loop invariant, lines 3–10 set
\[(T[j], T[j+1]) = \left( q^{\frac{L[i+j-1]+1}{2}}T_{L[i+j]}(q^{-\frac{1}{2}}A), q^{\frac{L[i+j-1]-1}{2}}T_{L[i+j]}(q^{-\frac{1}{2}}A) \right).\]

Hence line 23 makes
\[T[0] = q^{\frac{L[0]+1}{2}}T_{L[0]}(q^{-\frac{1}{2}}A)T_{L[0]}(q^{-\frac{1}{2}}A) - Q[0]A \]
\[= q^{\frac{L[0]+1}{2}} \left( T_{L[0]}(q^{-\frac{1}{2}}A)T_{L[0]}(q^{-\frac{1}{2}}A) - q^{-\frac{1}{2}}A \right) \text{ (by Lemma 2.2)} \]
\[= q^{\frac{L[0]+1}{2}}T_{L[0]}(q^{-\frac{1}{2}}A) \text{ (by Equation (9)).} \]

Decrementing $i$ for the next iteration, the loop invariant is preserved. The lemma follows.

We are now ready to prove the correctness of NGC.

**Theorem 2.4.** NGC($A, k$) returns the number $N_k$.

**Proof.** At termination of the for loop of lines 7–24 the value $i = 0$. Since the input $k$ is stored in $L[0]$, it follows from Lemma 2.3 that
\[T[0] = q^{\frac{1}{2}}T_k(q^{-\frac{1}{2}}A)\]
after the for loop of lines 7–24. Therefore lines 25–27 return the number $N_k$ by (8). The correctness follows.

**Theorem 2.5.** NGC($A, k$) runs in $O(n^\omega \log k)$ time.

**Proof.** Let $b_{\lceil \log k \rceil} b_{\lfloor \log k \rceil - 1} \cdots b_0$ denote the binary representation of $k$. Let $h$ denote the rightmost index with $b_h = 1$. Observe that lines 2–5 of INDICES($k$) increase the length of $L$ by $h + 1$ and lines 6–11 of INDICES($k$) increase the length of $L$ by double of $\lfloor \log k \rfloor - h$. Hence line 4 of NGC($A, k$) sets
\[l = 2 \lfloor \log k \rfloor - h + 1.\]
Since $n \times n$ matrix multiplication appears in NGC($A, k$) exactly $l - 1$ times, the time complexity is $O(n^\omega \log k)$.

Recall the number sequence $\{H_k\}_{k=1}^\infty$ from (1). To compute $H_k$, one may modify lines 25–27 of NCG($A, k$) to read as follows:
\[25 \text{ if } L[0] \text{ is odd} \]
\[26 \quad Q[0] = q^\frac{1}{2}Q[0] \]
\[27 \text{ return } 2(n-1) + Q[0] + Q[0]^{-1} - Q[0]^{-1} \text{trace}(T[0]) \]
Let HSEQUENCE($A, k$) denote the resulting procedure in the rest of this paper.

**Theorem 2.6.** HSEQUENCE($A, k$) runs in $O(n^\omega \log k)$ time.

**Proof.** Immediate from Theorem 2.5.

3. A connection between $\mu(X)$ and $\{H_k\}_{k=1}^\infty$

**Lemma 3.1.** For any integer $k \geq 1$ the following equation holds:
\[H_k = 2(n - 1) - \sum_{i=2}^{n} T_k(q^{-\frac{1}{2}}\lambda_i).\]

**Proof.** This lemma follows by substituting (8) into (1).

**Lemma 3.2.** If $s$ is a real number with $|s| \leq 2$ then $|T_k(s)| \leq 2$ for all integers $k \geq 0$. 

HUA-WEN HUANG
Proof. Recall that $\frac{1}{2}T_k(2x)$ is the $k$th Chebyshev polynomial of the first kind for all integers $k \geq 0$. Hence

$$T_k(2\cos \theta) = 2\cos k\theta \quad \text{for all real numbers } \theta.$$ 

The lemma follows from the above equality. \qed

Lemma 3.3 (Proposition 4.7, [12]). Let $S$ denote a nonempty finite multiset consisting of real numbers. If there is an even integer $k \geq 2$ with

\begin{equation}
\frac{1}{2|S|} \sum_{s \in S} T_k(s) \leq 1
\end{equation}

then $|s| \leq 1 + \frac{k}{4|S| - 3}$ for all $s \in S$.

We are ready to prove the following result:

Theorem 3.4. (i) If there exists an even integer $k \geq 2$ with $H_k \geq 0$ then

$$\mu(X) \leq 1 + \sqrt[4]{4n - 7}.$$ 

(ii) $H_k \geq 0$ for all integers $k \geq 2$ if and only if

$$\mu(X) \leq 2.$$ 

(iii) If $\mu(X) > 2$ then

$$\mu(X) = \lim_{k \to \infty} \sqrt{\frac{H_{2k+2}}{H_{2k}}} + \sqrt{\frac{H_{2k}}{H_{2k+2}}}.$$ 

Proof. (i): Suppose that there exists an even integer $k \geq 2$ with $H_k \geq 0$. By Lemma 3.1 it is equivalent to

$$\frac{1}{2(n-1)} \sum_{i=2}^{n} T_k(q^{-\frac{1}{2}}\lambda_i) \leq 1.$$ 

Applying Lemma 3.3 it follows that $q^{-\frac{1}{2}}|\lambda_i| \leq 1 + \sqrt[4]{4n - 7}$ for all $i = 2, 3, \ldots, n$. Therefore (i) follows.

(ii): ($\Rightarrow$) Since $H_k \geq 0$ for all even integers $k \geq 2$, it follows from Theorem 3.4(i) that

$$\mu(X) \leq \lim_{k \to \infty} 1 + \sqrt[4]{4n - 7} = 2.$$ 

($\Leftarrow$) Since $\mu(X) \leq 2$ it follows that $q^{-\frac{1}{2}}|\lambda_i| \leq 2$ for all $i = 2, 3, \ldots, n$. Combined with Lemma 3.2 this yields that $|T_k(q^{-\frac{1}{2}}\lambda_i)| \leq 2$ for all $i = 2, 3, \ldots, n$ and all integers $k \geq 0$. Hence $H_k \geq 0$ for all $k \geq 1$ by Lemma 3.1.

(iii): Recall that $\alpha_i$ is a root of $u^2 - q^{-\frac{1}{2}}\lambda_i u + 1$ for all $i = 1, 2, \ldots, n$. Observe that if $q^{-\frac{1}{2}}|\lambda_i| \leq 2$ then $|\alpha_i| = 1$. Since $\mu(X) > 2$ there is an $i \in \{2, 3, \ldots, n\}$ with $q^{-\frac{1}{2}}|\lambda_i| > 2$ and hence $\alpha_i$ is a real number with $\alpha_i \neq \pm 1$. Therefore

$$\beta = \max_{2 \leq i \leq n} \{|\alpha_i|, |\alpha_i|^{-1}\} > 1.$$ 

Since $f(x) = x + x^{-1}$ is strictly increasing on $(1, \infty)$, we have

$$\beta + \beta^{-1} = \max_{2 \leq i \leq n} |\alpha_i| + |\alpha_i|^{-1} = \mu(X).$$

By (7) we have

$$\lim_{k \to \infty} \frac{T_{2k}(\alpha_i + \alpha_i^{-1})}{\beta^{2k}} = \lim_{k \to \infty} \frac{\alpha_i^{2k} + \alpha_i^{-2k}}{\beta^{2k}} = \begin{cases} 1 & \text{if } \beta = \max\{|\alpha_i|, |\alpha_i|^{-1}\}, \\ 0 & \text{else} \end{cases}$$
for all $i = 2, 3, \ldots, n$. Combined with Lemma 3.1 this yields that $H_{2k}$ is asymptotic to $-m\beta^{2k}$ as $k$ approaches to $\infty$ where $m$ is the number of $i \in \{2, 3, \ldots, n\}$ with $\beta = \max\{|\alpha_i|, |\alpha_i|^{-1}\}$. By the above comments the statement (iii) follows. 

4. A FAST ALGORITHM TO EVALUATE $\mu(X)$

With the inputs $A$ and a real number $\varepsilon > 0$, Theorem 3.4 yields the following algorithm that return TRUE if $\mu(X)$ or its estimation is less than or equal to $2 + \varepsilon$ and return FALSE else:

```plaintext
SpectralExpansion(A, \varepsilon)
1   n = the number of rows of A
2   k = 2[log(4n - 7)/2 log(1 + \varepsilon)]
3   k' = k + 2
4   H = Hsequence(A, k)
5   H' = Hsequence(A, k')
6   if $H \geq 0$ or $H' \geq 0$
7       return TRUE
8   else \lambda = \sqrt{H'/H} + \sqrt{H/H'}
9       print \lambda “ is an estimation of $\mu(X)$.”
10      if $\lambda \leq 2 + \varepsilon$
11         return TRUE
12      else return FALSE
```

**Theorem 4.1.** $\text{SpectralExpansion}(A, \varepsilon)$ runs in $O(n^\omega \log \log 1 + \varepsilon n)$ time.

**Proof.** By line 2 the value $k = 2[\log(4n - 7)/2 \log(1 + \varepsilon)] = O(\log 1 + \varepsilon n)$. Hence the lines 4 and 5 take $O(n^\omega \log \log 1 + \varepsilon n)$ time by Theorem 2.6. The other lines run in $O(1)$ time. The result follows. 

We have a remark on the procedure $\text{SpectralExpansion}$. In the case of $H < 0$ and $H' < 0$, if the parameter $\varepsilon$ is small enough $\text{SpectralExpansion}(A, \varepsilon)$ gives a nice estimation of $\mu(X)$ theoretically; however, if $\varepsilon$ is not small enough the procedure $\text{SpectralExpansion}(A, \varepsilon)$ perhaps returns incorrect information.

**Acknowledgements.** The research is supported by the Ministry of Science and Technology of Taiwan under the project MOST 106-2628-M-008-001-MY4.

**References**

1. N. Alon, *Eigenvalues and expanders*, Combinatorica 6 (1986), 83–96.
2. N. Alon and F. R. K. Chung, *Explicit construction of linear sized tolerant networks*, Discrete Mathematics 72 (1988), 15–19.
3. H. Bass, *The Ihara–Selberg zeta function of a tree lattice*, International Journal of Mathematics 6 (1992), 717–797.
4. D. Coppersmith and S. Winograd, *Matrix multiplication via arithmetic progressions*, Journal of Symbolic Computation 9 (1990), 251–280.
5. G. Davidoff, P. Sarnak, and A. Valette, *Elementary number theory, group theory and Ramanujan graphs*, Cambridge University Press, Cambridge, 2003.
6. J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
7. J. Dodziuk, *Difference equations, isoperimetric inequality and transience of certain random walks*, Transactions of the American Mathematical Society 284 (1984), 787–794.
8. J. Friedman, *Some geometric aspects of graphs and their eigenfunctions*, Duke Mathematical Journal 69 (1993), 487–525.
9. F. Le Gall, *Powers of tensors and fast matrix multiplication*, Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation (2014), 296–303.
10. K. Hashimoto, *Zeta functions of finite graphs and representations of p-adic groups*, Automorphic forms and geometry of arithmetic varieties, Advanced Studies in Pure Mathematics 15 (1989), 211–280.
11. S. Hoory, N. Linial, and A. Wigderson, *Expander graphs and their applications*, Bulletin of the American Mathematical Society 43 (2006), 439–561.
12. H.-W. Huang, *Ihara Zeta function, coefficients of Maclaurin series and Ramanujan graphs*, International Journal of Mathematics 31 (2020), 2050082 (10 pages).
13. H.-W. Huang and W.-C. W. Li, *Unramified graph covers of finite degree*, Connections in discrete mathematics: A celebration of the work of Ron Graham (Cambridge, United Kingdom) (S. Butler, J. Cooper, and G. Hurlbert, eds.), Cambridge University Press, 2018, pp. 104–124.
14. Y. Ihara, *On discrete subgroups of the two by two projective linear group over p-adic fields*, Journal of the Mathematical Society of Japan 18 (1966), 219–235.
15. W.-C. W. Li, *Zeta and L-functions in number theory and combinatorics*, Regional Conference Series in Mathematics 129, American Mathematical Society, Providence, RI, 2019.
16. A. Lubotzky, *Discrete groups, expanding graphs and invariant measures*, Birkhäuser, Basel, 1994.
17. A. Lubotzky, R. Phillips, and P. Sarnak, *Ramanujan graphs*, Combinatorica 8 (1988), 261–277.
18. A. Nilli, *On the second eigenvalue of a graph*, Discrete Mathematics 91 (1991), 207–210.
19. J.-P. Serre, *Trees*, Springer-Verlag Berlin Heidelberg, 1980, translated from the French by John Stillwell.
20. T. Sunada, *L-functions in geometry and some applications*, Curvature and Topology of Riemannian Manifolds (Berlin) (K. Shiohama, T. Sakai, and T. Sunada, eds.), Lecture Notes in Mathematics 1201, Springer, 1986, pp. 266–284.
21. A. A. Terras, *Zeta functions of graphs: A stroll through the garden*, Cambridge Studies in Advanced Mathematics 128, Cambridge University Press, Cambridge, UK; New York, 2010.
22. V. V. Williams, *Multiplying matrices faster than coppersmith-winograd*, Proceedings of the 44th Annual ACM Symposium on Theory of Computing (2012), 887–898.

Department of Mathematics, National Central University, Chung-Li 32001 Taiwan
Email address: hauwenh@math.ncu.edu.tw