# Expander Graphs
# Exercise Sheet 3

**Question 1.** By choosing an appropriate $t = \Theta(\log n)$, $m = \text{poly}(n)$ and $\epsilon = \Theta(1)$, show that the algorithm $B'$ described the notes demonstrates a reduction of the problem in Theorem 3.13 to the problem in Theorem 3.13.

**Question 2.** Given a matrix $A$ with eigenvalues $\lambda_1 \geq \ldots \geq \lambda_n$ show that the *Rayleigh quotient* of any vector $\boldsymbol{x}$ satisfies

$$\frac{\boldsymbol{x}A\boldsymbol{x}^T}{\boldsymbol{x}\boldsymbol{x}^T} \in [\lambda_n, \lambda_1].$$

Show that for every graph $G$ the Laplacian $L(G)$ is positive semi-definite. and satisfies:

- $L = L_G = dI - A(G)$;
- The spectrum of $L$ is in $[0, 2d]$.
- The smallest eigenvalue of $L$ is zero.
- The spectral gap of $G$ is equal to the smallest positive eigenvalue of $L$.

**Question 3** (Matrix-tree theorem)**.** Let $E_i$ be a matrix such that $E_{i,i} = 1$ and $E_{j,k} = 0$ otherwise. Let $A$ be an abitrary matrix and let $A[i]$ be the matrix obtained by deleting the $i$th row and column of $A$. Show that $\det(A + E_i) = \det(A) + \det(A[i])$.

Show that the number of spanning trees of a graph $G$ is given by $\det(L(G)[i])$ for any $i$.

(* Conclude that the number of spanning trees of $G$ is given by $\frac{1}{n}\prod_{i=1}^{n-1}\lambda_i$, where $\lambda_i$ are the eigenvalues of $L(G)$.)

**Question 4.** Let $G$ be an $(n, d, \alpha)$-graph and let $\rho > 0$. Show that for every subset $S \subseteq V(G)$ of size $|S| \leq \rho n$

$$|\Gamma(S)| \geq \frac{1}{\rho(1-\alpha^2) + \alpha^2}|S|,$$

where $\Gamma(S)$ is the inclusive neighbourhood of $S$.

(Hint : Consider $||\hat{A}\mathbf{1}_S||_2^2$)

**Question 5.** Let $d \geq 3$ and let $\delta > 0$. Show that there exists an $\epsilon > 0$ such that for almost every $(n, d)$-graph $G$, every subset $S \subseteq V(G)$ with $|S| \leq \epsilon$ satisfies

$$|\Gamma(S)| \geq d - 1 - \delta|S|.$$

(Hint: This can be obtained from the proof of Theorem 4.7. In particular no more probabilistic statements need proving)