Midterm 1

Printed Name:

Due date:

1. The exam is due Monday 26 Sep at 8AM.
2. You may submit it either on sakai electronically or by putting a physical copy under my door.

Rules:

1. The exam is untimed. You do not have to complete the exam in a single sitting. You may pause and restart whenever you'd like.
2. You may use any non-human resources that you like, including notes, books, internet articles, and computers.
3. You are not allowed to discuss the exam in any way with any human until after the due date. This includes:
   (a) obviously bad behavior like copying answers,
   (b) more banal behavior such as:
      i. telling your friend "Problem 6 was really hard" or
      ii. asking your friend "Have you completed the exam yet?"

   Even after you finish the exam, you may not discuss it.
4. If you do have questions about the exam, you should email me the questions rather than posting to github.

Grading:

1. For the True/False/Open questions: Each correct answer will be awarded +1 point, each incorrect answer will result in a -1 point penalty, and each blank answer will result in 0 points.
2. All other problems are worth 1 point, with no penalty for incorrect answers.
3. There are 16 points possible on the exam. Your final grade entered into sakai will be
$$\min\{15, \text{the number of points earned}\}.$$
4. If you find a substantive error on the exam, then I will award you +1 bonus point.

Good luck :)

Problem 1. For each statement below, circle `True` if the statement is known to be true, `False` if the statement is known to be false, and `Open` if the statement is not known to be either true or false. Ensure that you pay careful attention to the formal definitions of asymptotic notation in your responses.

1. `True` `False` `Open` Let A and B be dense $n \times n$ matrices. Then the fastest possible algorithm for computing the matrix product AB has runtime $\Omega(n^2 \log^2 n)$.

2. `True` `False` `Open` Let A be a dense $n \times n$ matrix. Then the fastest possible algorithm for computing the matrix product A^3 has runtime $O(n^3)$.

3. `True` `False` `Open` Let $\mathbf{x}$ be a dense n-dimensional vector. Then the fastest possible algorithm for computing the expression $\mathbf{x}\mathbf{x}^T\mathbf{x}$ has runtime $\Theta(n)$.

4. ~~`True` `False` `Open` It is always true that~~ $\cancel{\|\bar{\mathbf{P}}\|_2 = \|\bar{\bar{\mathbf{P}}}\|_2}$.

   (Everyone awarded +1 point for this problem, regardless of your answer. It's true that the top eigenvalues for both matrices must be equal to 1 because they are both stochastic, but the L2 norm technically measures the top singular value rather than the top eigenvalue, and these need not be equal.)

5. `True` `False` `Open` Assume we are computing the pagerank of a graph with 10 nodes using $\alpha = 0.85$. Then it is possible for the L1 norm of the pagerank vector to be equal to 1.

6. `True` `False` `Open` There exists a valid $\mathbf{P}$ matrix such that $\text{nnz}(\mathbf{P}) = n$ and $\text{nnz}(\mathbf{PP}) = n$.

7. `True` `False` `Open` Every stochastic matrix is irreducible.

8. `True` `False` `Open` $\text{nnz}(\bar{\bar{\mathbf{P}}}) = O(n^3)$.

9. `True` `False` `Open` It is possible for the $\bar{\mathbf{P}}$ matrix to be primitive.

10. `True` `False` `Open` It is possible for the $\bar{\bar{\mathbf{P}}}$ matrix to have an eigenvalue λ that is greater than α but less than 1; that is, an eigenvalue satisfying $1 > \lambda > \alpha$.

Problem 2. Either prove or give a counterexample to the following claim: For any n-dimensional vector $\mathbf{x}$, it is true that $\|\mathbf{x}^T\bar{\bar{\mathbf{P}}}\|_2 \leq \|\mathbf{x}\|_2$.

Solution: The statement is FALSE, and I awarded full credit for any valid counterexample. For example,

$$\mathbf{x}^T = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \qquad \text{and} \qquad \bar{\bar{\mathbf{P}}} = \begin{pmatrix} 0.5 & 0.25 & 0.25 \\ 0.5 & 0.25 & 0.25 \\ 0.5 & 0.25 & 0.25 \end{pmatrix} \tag{1}$$

results in $\|\mathbf{x}\|_2 = \sqrt{3} \approx 1.732$ and $\|\mathbf{x}^T\bar{\bar{\mathbf{P}}}\|_2 \approx 1.837$.

Solution: I also awarded full credit if you claimed that the answer was TRUE somehow due to $\bar{\bar{\mathbf{P}}}$ having a maximum eigenvalue of 1. That is, if you stated something like:

$$\|\mathbf{x}^T\bar{\bar{\mathbf{P}}}\|_2 \leq \|\mathbf{x}\|_2\|\bar{\bar{\mathbf{P}}}\|_2 \tag{2}$$

$$= \|\mathbf{x}\|_2 \tag{3}$$

where the first step is true due to the properties of norms and the second step is true because $\|\bar{\bar{\mathbf{P}}}\|_2 = 1$.

Unfortunately, it is not true that $\|\bar{\bar{\mathbf{P}}}\|_2 = 1$. I misspoke in class when I said that the L2 norm of a matrix equals its top eigenvalue; instead, it equals it's top *singular value*. The singular values of a matrix A are defined to be the square root of the eigenvalues of A^TA. For real, symmetric matrices, the singular values and the eigenvalues are always the same; otherwise, they do not have to be the same (as in the case of the $\bar{\bar{\mathbf{P}}}$ example above.

If you said the answer was TRUE, and it looked to me like your justification was something like this, but you didn't explicitly state anywhere that the max eigenvalue is 1 or that $\|\bar{\bar{\mathbf{P}}}\|_2 = 1$, then you got 0.5 points.

Problem 3. Recall that Equation (5.1) states that

$$\mathbf{x}^{(k)T} = \alpha \mathbf{x}^{(k-1)T}\mathbf{P} + \big(\alpha \mathbf{x}^{(k-1)T}\mathbf{a} + (1-\alpha)\big)\mathbf{v}^T.$$

In class, we computed the runtime for a single iteration of Equation (5.1) assuming that $\mathbf{P}$ is sparse. What is the runtime of computing a single iteration if $\mathbf{P}$ is dense?

Solution: $O(n^2)$

If you didn't simplify the solution, you received +0.5

Problem 4. If $\mathbf{P}$ is dense (as in the previous problem), then what is the overall runtime for using the power method to compute an approximate pagerank vector with residual accuracy ϵ?

Solution: $O\left(\frac{\log \epsilon}{\log \alpha} n^2\right)$

If you didn't simplify the solution, you received +0.5

If you used the incorrect value from the previous solution, then you received +0.5

Problem 5. In practice, no one uses the EAPM to compute pagerank vectors. Why?

Solution: The runtime for the EAPM is $O\big((\log \frac{\log \epsilon}{\log \alpha})n^3\big)$ compared to $O\big(\frac{\log \epsilon}{\log \alpha}(\mathrm{nnz}(P) + n)\big)$ for the standard PM. The EAPM has better dependencies on ϵ, but much worse dependencies on n. In real-world problems, ϵ is always less than about 10^{-6}, but n can grow extremely large. Therefore, we care about the dependence on n much more than the dependence on ϵ.

Problem 6. Let A be an $n \times n$ dense matrix and k be a positive integer. Describe a procedure for computing A^k in time $O(n^3 \log k)$.

Solution: In class, we used the recursive formula

$$\mathbf{Q}_k = \begin{cases} \bar{\bar{\mathbf{P}}} & \text{if } k = 0 \\ \mathbf{Q}_{k-1}\mathbf{Q}_{k-1} & \text{otherwise} \end{cases} \tag{4}$$

to compute the $\bar{\bar{\mathbf{P}}}^k$ expression quickly in the EAPM, and any variation on this earned full credit.

Technically, the procedure above only works for values of k that are powers of 2, and the question is asking for a procedure that works for any k. The following recursion computes A^k for all values of k:

$$A^k = \begin{cases} 1 & \text{if } k = 0 \\ A \cdot (A^2)^{(k-1)/2} & \text{if } k \text{ is odd} \\ (A^2)^{k/2} & \text{if } k \text{ is even} \end{cases} . \tag{5}$$

Problem 7. Describe a problem that can be solved using pagerank. (This is similar to Problem 1 from the Pagerank III notes, but your example problem should be different than the ones we did in class.)

Solution: Any remotely reasonable solution earned full credit.