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 $\|\bar{\mathbf{P}}\|_2 = \|\bar{\bar{\mathbf{P}}}\|_2$.

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 $\text{nnz}(\mathbf{P}\mathbf{P}) = n$.

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

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?

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?

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

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 $O(n^3 \log k)$.

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.)