CS246: Mining Massive Data Sets **Winter 2024**

Exam

- Your Name: ____
- Your SUNetID: __________________________________
- Your SUID: __________________________________

I acknowledge and accept the Stanford Honor Code.

Signature: ____

1. There are 13 questions in this exam (numbered 2 to 14); the maximum score that you can obtain is 180 points. These questions require thought but do not require long answers. Please be as concise as possible. You can use the number of points as a rough estimate of how long we think a question may take you. Don't worry: we don't expect you to complete all the questions.
2. This exam is open-book and open-notes. You may use notes (digitally created notes are allowed) and/or lecture slides and/or any reference material. However, **answers should be written in your own words**.
3. Acceptable uses of computer:
   - You may access the Internet, but you may not communicate with any other person. Similarly, AI-driven code completion tools including ChatGPT and GitHub Copilot are not allowed
   - You may use your computer to write code or do any scientific computation, though writing code is not required to solve any of the problems in this exam.
   - You can use your computer as a calculator or an e-reader.
4. Collaboration with other students is not allowed in any form. Please do not discuss the exam with anyone until after grades are released.
5. If you have any clarifying questions, make a **private post on Ed**. It is very important that your post is private; if it is public, we may deduct points from your exam grade.
6. Please submit your answers here on Gradescope. You have three options to submit your answers: (1) to upload one file for all questions (at the top of the Gradescope exam) (2) to upload one file per question, in a file upload field in the last sub-question; or (3) to write your answers directly in the text fields in the sub-questions.
7. Numerical answers may be left as fractions, as decimals rounded to **2 decimal places**, or as radicals (e.g., $\sqrt{2}$).

8. The course (and thus your score on the exam) will be curved. Do not stress if you are unable to finish every question in time.

9. **Please save your responses in a separate document as Gradescope does not automatically save partially completed submissions**

2 True/False Questions (15 points)

For each of the statements given below, mark them as either True or False. For full credit, also add a 1-2 line reasoning behind your answer.

1. **(1.5 point)** If ABC is a frequent itemset and BCD is not a frequent itemset, then ABD cannot be a frequent itemset
2. **(1.5 point)** Increasing the number of hash tables in LSH decreases the probability of false positives.
3. **(1.5 point)** K-means clustering can handle outliers well by assigning them to their own clusters.
4. **(1.5 point)** Pruning is a technique that can help combat decision trees from overfitting.
5. **(1.5 point)** For the attention mechanism in a GAT layer, the model learns a single attention weight for each node.
6. **(1.5 point)** A group of bloggers decides to cross-promote by linking each other's websites on all of their blogs. This is a spider trap because if we represent each website as a node, the bloggers' websites will form a loop.
7. **(1.5 point)** Spam mass will always be non-negative for all pages in a graph because it measures the percentage of page rank that comes from spam.
8. **(1.5 point)** Embeddings learned through neural networks are always linearly separable in the embedding space.
9. **(1.5 point)** The information gain at the root node of the decision tree is greater than or equal to the information gain at any other node.
10. **(1.5 point)** The greedy algorithm has a higher competitive ratio than generalized version of BALANCE algorithm.

3 Locality Sensitive Hashing (17 points)

You would like to use shingling and locality sensitive hashing to identify possible plagiarism in student essays. There are two methods you would like to try:

- Compare an essay P with a publication Q in your database using shingling and Jaccard similarity.
- Measure similarity with cosine similarity, where the shingle sets are viewed as binary vectors

In the following questions, consider an essay P with 1500 unique length-c shingles in it and a publication Q with 4500 unique length-c shingles. Let S_P and S_Q be the shingle sets for P and Q respectively, so $|S_P| = 1500$ and $|S_Q| = 4500$, and assume that P shares 1000 shingles with Q.

1. **(6 points)** What is the Jaccard distance between the shingle sets of the two documents? Also calculate $\Pr(\text{MinHash}(S_P) = \text{MinHash}(S_Q))$.
2. **(8 points)** Let m be the number of all possible length-c shingles (i.e., all ordered sets of c length words in the English language). Represent S_P and S_Q by length m binary vectors x_P and x_Q, with a 1 in every position corresponding to a shingle they contain.
   - What is the cosine similarity between x_P and x_Q (again assuming P was fully copied from a portion Q)?
   - What is $\Pr\left(\text{SimHash}\left(x_P\right) = \text{SimHash}\left(x_Q\right)\right)$?
     Hint: Use that for any two vectors u, v, $\Pr(\text{SimHash}(u) = \text{SimHash}(v)) = 1 - \frac{\theta}{\pi}$ where θ is the angle between u and v in radians.
3. **(3 points)** Given the above, which similarity metric and hash function would you pick for the plagiarism detection task and why?

4 Clustering (12 points)

For this problem, please use Euclidean distance to cluster. In case of ties, please assign a point to cluster with the lowest centroid norm.

1. **(4 points)** In hierarchical clustering, the key operation is to repeatedly combine the two nearest clusters. Let us consider a one-dimensional space (R for example). We wish to perform a hierarchical clustering of points 2, 5, 10, 14, 17, and 26. Show **what happens at each step** until there are two clusters, and **give these two final clusters**.

2. **(4 points)** Now, we would like to perform K-means clustering on the same points with the initial centroids being 5, 17. Show **what happens at each step** until there are two clusters, and **give these two final clusters**

3. **(4 points)** If the goal is to cluster the data points in Figure 1 to 2 clusters using Euclidean distance, which algorithm(s) works well? Select all that applies for each of them; you don't need to provide justifications. Assume that images are made to scale and different colors represent cluster assignments.

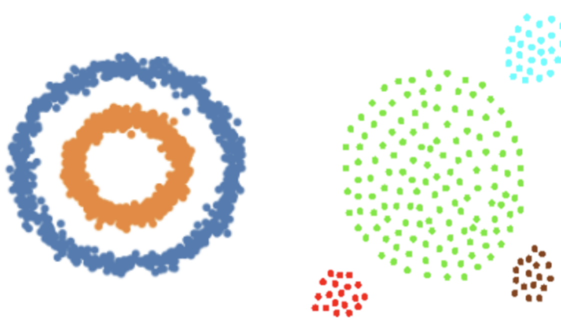


Figure 1: Datasets to cluster

(a) Left: K-means/ Hierarchical clustering/ None/ Both

(b) Right: K-means/ Hierarchical clustering/ None/ Both

5 Dimensionality Reduction (13 points)

1. **(3 points) Low-Rank Approximation Using SVD.** Consider the following 2×3 matrix: $A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$. You are given its (thin) Singular Value Decomposition (SVD) as the following:
$$A = U\Sigma V^\top = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} \sqrt{3} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{pmatrix}.$$
Find the rank-1 matrix $B \in \mathbb{R}^{2\times 3}$ which minimizes $\|A - B\|_F$, i.e. the solution to the optimization problem
$$\min_{\text{rank}(B)=1} \|A - B\|_F\,.$$

2. **(5 points) The Moore-Penrose Pseudoinverse.** Recall that in the discussion of CUR decomposition in the lecture, we have introduced the notion of the pseudoinverse A^+ of a matrix $A \in \mathbb{R}^{m\times n}$ with singular value decomposition $A = U\Sigma V^\top$ as $A^+ = V\Sigma^+ U^\top$. Here, denote $r = \text{rank}(A) \le \min\{m, n\} = k$, and for $\Sigma = \text{diag}(\sigma_1, \cdots, \sigma_k)$, where $\sigma_1 \ge \cdots \ge \sigma_r > \sigma_{r+1} = \cdots = \sigma_k = 0$, Σ^+ is defined as $\Sigma^+ = \text{diag}(\sigma_1^{-1}, \cdots, \sigma_r^{-1}, 0, \cdots, 0)$.

   However, the idea of a pseudoinverse was introduced as a generalization of matrix inverse, which requires it to satisfy the four following Moore-Penrose conditions:
$$AA^+A = A, \quad A^+AA^+ = A^+, \quad (AA^+)^\top = AA^+, \quad (A^+A)^\top = A^+A.$$
   Penrose (1955) noticed that the matrix satisfying the four conditions is unique (you don't need to prove this). Prove that $A^+ = V\Sigma^+U^\top$ satisfies these conditions, which implies that our definition is a equivalent one. Furthermore, prove that in the case of A having full column rank, the pseudoinverse A^+ has the following familiar form:
$$A^+ = (A^\top A)^{-1}A^\top.$$

3. **(5 points) Random Matrix Multiplication.** Out of the same sampling idea as in the CUR decomposition algorithm, we have the following algorithm for approximating the product of two matrices:
   Input: $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times p}$, sample size $c \in \mathbb{N}$ and sampling distribution parameterized by $\{p_i\}_{i=1}^n$ such that $p_i \ge 0$ and $\sum_{i=1}^n p_i = 1$.
   Output: $C \in \mathbb{R}^{m\times c}$ and $R \in \mathbb{R}^{c\times p}$ such that $CR \approx AB$.

   (a) For $t = 1$ to c,
      i. Sample $i_t \in \{1, \cdots, n\}$ with $\mathbb{P}(i_t = k) = p_k$, $k = 1, \cdots, n$, indepedently and with replacement;
      ii. Set $C^{(t)} = A^{(i_t)}/\sqrt{cp_{i_t}}$ and $R_{(t)} = B_{(i_t)}/\sqrt{cp_{i_t}}$, where the notations $X^{(i)}$ and $Y_{(j)}$ mean the i-th column of X and the j-th row of Y, respectively.

   (b) Return C, R.

   Prove that the matrix product CR is an approximation of AB in the sense of expectation, i.e.
$$\mathbb{E}[CR] = AB.$$

6 Recommender Systems (12 points)

Suppose you have a database of books and users. The database stores each book's length (in thousands of words), genre, and author and each user's rating information for the books they have read: 1 if they liked it, 0 if they did not.

The table below summarizes the database:

Book	Length	Genre	Author	Total number of ratings
A	72k	Sci-fi	A1	100
B	108k	Sci-fi	A1	1000
C	383k	Fantasy	A2	600
D	40k	Mystery	A3	40
E	52k	Mystery	A3	8
F	12k	Horror	A4	200

Consider user U1, a longtime user of your database who is interested in the long fantasy novels. You have a recommender system R that suggested the book B to user U1. R could be one or more of the following options:

- User-user collaborative filtering
- Item-item collaborative filtering
- Content-based recommender system

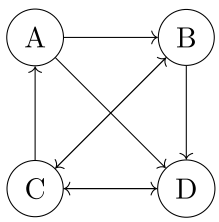
Answer the following questions:

1. **(4 points)** Which option(s) do you think R could be? Explain your answer. (If more than one option is possible, you need to state them all.)
2. **(4 points)** Assume R is a content-based recommender system. What book would R recommend to a new user U2 if they want to read science fiction?
3. **(4 points)** Item-item collaborative filtering is seen to work better than user-user because users have multiple tastes. But this also means that users like to be recommended a variety of books.

   Given the genre of each book (sci-fi, fantasy, mystery, horror) and an item-item collaborative filtering recommender system that predicts k top-books to a user (k can be an input to the recommender), suggest a way to find top 3 books to a user such that the recommender will try to incorporate books from different genres as well. (Note: Explain in 3–5 lines maximum, no rigorous proof is required.)

7 PageRank (14 points)

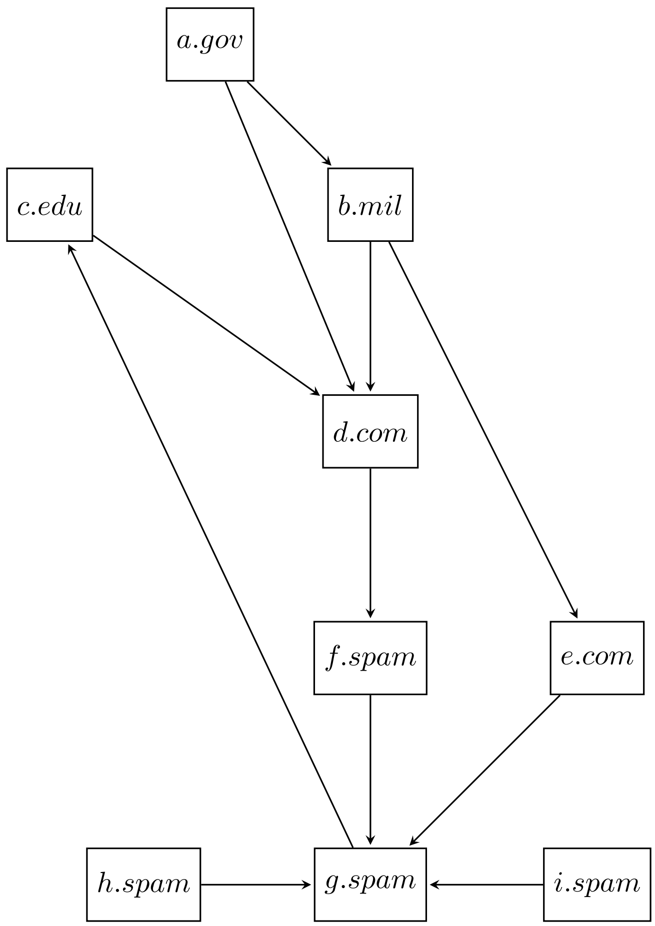
1. **(3 points)** How can teleportation be useful in personalized PageRank?
2. **(3 points)** In an alternate version of Pagerank, say we consider outgoing edges in calculating the importance of a node instead of incoming edges. Describe the potential flaws of this approach.
3. **(4 points)** Construct a graph of 4 nodes where each node has the same PageRank no matter what the teleportation probability is set to. Justify your answer in 1-2 lines.
4. **(4 points)** What is the column-stochastic adjacency matrix M corresponding to the below graph?



8 Extended PageRank (15 points)

In your first week as a new intern at a search engine startup, you are tasked with handling spam links.

1. **(7 points)** Your manager tells you to use the Trusted Propagation model to deal with spam where each website node will have a trust value associated with it and trusts below a threshold will be eliminated as spam before we calculate PageRank. Your manager wants you to use the trusted domain strategy where the trust of .gov, .edu and .mil pages are set to 1. Given a trust threshold of 0.18, what is the maximum value of β that will ensure that ALL spam nodes in the below graph are flagged correctly? Show your work.



2. **(4 points)** Your coworker tells you that spam pages cannot get high page ranks and picking the top k pages as the seed set is better than using trusted domains. For the above graph, would this approach work well? Justify your answer.

3. **(4 points)** The startup decides to pivot to building an image recommendation system. Your manager tells you to copy Pixie, Pinterest's recommendation algorithm, and to minimize execution costs as much as possible. You know that Pixie uses early stopping to reduce the number of steps in random walk simulations. Pixie achieves this by performing the walk until the 1000$^{\text{th}}$ most visited pin has at least 20 visits. What could happen if you set a low value for the minimum threshold of visit counts for early stopping?

Hint: Consider what will happen if you make an algorithm simulate random walks on a bipartite graph with more than 1000 nodes and a density greater than 5 and stops when the 100th most visited pin/node has at least 2 visits.

9 Community Detection (12 points)

1. **(6 points)** Your friend is interested in the Personalized Page Rank Algorithms, first **derive the worst-case time complexity of the algorithm**, and then **explain why it is necessary to first sort the nodes before sweeping**. Recall that the PPR algorithm is made up of 4 steps: picking a set s of seed nodes, running PPR with teleport=s, sorting the nodes by the PPR score, and sweeping over the nodes to identify clusters. **Be sure to analyze the complexity of each stage**. Express the final answer in terms of $|s|$ the cardinality of set s, $|V|$ the number of nodes in the graph, and/or $|E|$ the number of edges in the graph. **You may assume that sorting can be done in linear time.**

2. **(3 points)** Recall the Louvain algorithm for community detection. Is the algorithm guaranteed to converge? If so, is it guaranteed to stop at the community assignment with maximum modularity? Explain why. If the algorithm is not guaranteed to stop, explain why not.

3. **(3 points)** Your friend did not like the PPR algorithm and Louvain algorithm. They think these algorithms are too difficult to understand. So, they proposed to use NN-Descent to do community detection. Do you think the NN-Descent algorithm can directly replace PPR or Louvain? Explain why or why not.

10 Frequent Itemsets (10 points)

We have five items (a, b, c, d, e), but only information on the first four (item e has no known information). The information we do have is presented below:

Itemset	Frequency
a	7
b	10
c	17
d	13
(a, b)	7
(b, d)	2
(b, c)	6

We will use a support threshold s of $s = 5$. Please categorize the following into whether or not the itemset is **guaranteed to be frequent**, **could be frequent of infrequent**, or **guaranteed to be infrequent**. ***Explain your reasoning.***

1. **(2 points)** (a, b, c, d)
2. **(2 points)** (a, b, d)
3. **(2 points)** (a, b, e)
4. **(2 points)** (a, d)
5. **(2 points)** (c, d)

11 Decision Trees (18 points)

Our goal is to create a decision tree for n training examples (x, y) where each point x consists of d features or attributes.

1. **(3 points) True or False**: Suppose we want the decision tree to predict the house price y based on its features x (e.g. size, number of rooms, etc.), where the price can be any positive real number. When building the tree, we would prefer to calculate the entropy and use information gain to decide which attribute to split on. Please provide a brief explanation.

2. **(3 points)** Assume that the dataset is self-consistent, meaning that for any two examples (x, y) and (x', y'), $x = x'$ implies $y = y'$. When learning a decision tree on this dataset, what is the best training accuracy you can achieve (and why) if we don't have any constraints on the decision tree? Does this imply anything that could be problematic in practice? Please provide a brief explanation.

3. **(9 points)** You are given the following dataset which contains 6 training examples. Let's consider a situation where we aim to employ three-way splits. In order to achieve this, we must select two values a, b such that $a < b$ and then partition the data into three sets based on the following criteria: $x^{(1)} < a; a \leq x^{(1)} < b; x^{(1)} \geq b$. Suppose we choose $a = 2.0$ and $b = 2.5$ at the root. What is the information gain of this split? Use the entropy function with log base 2 for your calculation.

$x^{(1)}$	y
1.3	Yes
1.7	No
2.1	No
2.7	Yes
3.0	Yes
4.0	No

4. **(3 points)** Use the same dataset and three-way splits strategy as in the previous part. Additionally, we aim to ensure that each branch of the split contains at least one vector. In order to find the split that has the highest information gain when splitting on $x^{(1)}$, how many $\{a, b\}$ pairs do we need to consider?

12 Mining Data Streams (12 points)

Suppose you are working for a social media company. The company collects a stream of user posts on various topics, such as sports, politics, entertainment, etc. You are given the task of counting how many distinct users have posted on a specified selection of topics, say P. The company policy is very strict towards these topics. They are okay with counting some users who have not posted about these topics but **NOT** okay with missing any user who has posted about them.

Formally, you are given a stream of <user,topic> tuples and a set P of topics. You know that the total number of users that have posted on the platform is $|U| = 100,000$.

Given your CS246 experience, you decide to apply a combination of Bloom Filters and Flajolet-Martin algorithm. Your new algorithm works as follows:

1. Pick a bloom filter on topics using a bit array of $|B|$ bits and k different hash functions.
2. Use the bloom filter to select tuples from the stream whose topic belongs to the set P, and remove the remaining tuples from the stream.
3. To the stream of selected tuples, apply the Flajolet-Martin algorithm over the users to count the number of unique users.

Based on the above algorithm, answer the following questions:

1. **(4 points)** You are first experimenting with the bloom filter to find the optimal k value which minimizes the false positive rate. So far, you have the rough measurements given in the following table.

k	False Positive Rate
3	0.95%
8	0.57%
12	0.32%
14	1.34%

   To conduct further experiments, which of the following range(s) of values should you focus your efforts on.

   (a) $k < 3$
   (b) $3 < k < 8$
   (c) $8 < k < 12$
   (d) $12 < k < 14$
   (e) $14 < k$

2. **(4 points)** Let's assume you figured out the optimal k and that gives you a false positive rate of 0.25%. Now you apply Flajolet-Martin algorithm. You note that the maximum number of trailing zeros in the binary strings you have seen so far is 12. Calculate the estimated number users who have posted about topics in P and show your calculations. You don't have to calculate the final value, feel free to leave the answer in expression form. (*Hint: Don't forget to take the bloom filtering into account. Remember that the total number of users that have posted on the platform is 100,000.*)

3. **(4 points)** Suppose that the company policy on topics in P has changed from very strict to very relaxed. Now, counting users who do not post about topics in P is **NOT** okay but missing some users who post about topics in P is fine. Does the algorithm you designed still work? Explain your reasoning.

13 Embeddings & GNN (15 points)

1. **(2 points)** One known problem with word embeddings is that antonyms, which are words with opposite meanings, often have similar embeddings. For example, by searching for most similar words to "increase" or "enter" using cosine similarity, words with clear antonyms like "decrease" or "exit" are likely to appear among the top similar words. Discuss in 1-2 sentences why embeddings might have this tendency.

2. **(2 points)** For undercomplete autoencoder, what is the relative size relationship between bottleneck dimension and input dimension? Why is such a relationship enforced?

3. **(1 point)** In random walk approaches for node embeddings, DeepWalk runs fixed-length, unbiased random walks starting from each node while node2vec uses flexible, biased random walks that can trade off between local and global views of the network. Node2vec has two parameters: return parameter p and in-out parameter q. What is the correponding numerical value of p and q if we see the sampling strategy in DeepWalk as a special case of node2vec? Please answer **p = ? and q = ?**

4. **(2 points)** For a graph G, each GNN layer performs O(___) message passing and performs O(___) neighbor aggregation.Please fill in the blanks using functions of $|V|$ and $|E|$.

5. **(8 points)** Prim's algorithm (also known as Jarník's algorithm) is a greedy algorithm that finds a minimum spanning tree for a weighted undirected graph. This means it finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized. The algorithm operates by building this tree one vertex at a time, from an arbitrary starting vertex, at each step adding the cheapest possible connection from the tree to another vertex (the distance between the partial spanning tree and a node outside of it is the shortest distance between any node in the tree and the node outside). In mathematical words, we add node $v = \text{argmin}_{x \notin T} d(x, T)$ where $d(x, T) = \min_{u \in T} d(x, u)$ and T is the partial spanning tree constructed from previous steps, to the spanning tree.

Algorithm 1: Prim's Algorithm

Data: A connected, undirected graph $G = (V, E)$ with weight function $w : E \to \mathbb{R}$ and a root vertex r.
Result: A minimum spanning tree of G.

```
key(v) ← ∞, ∀v ∈ V
key(r) ← 0
Q ← (key(v), v), ∀v ∈ V
p(v) ← NIL, ∀v ∈ V
A ← ∅
while Q ≠ ∅ do
    u ← ExtractMin(Q)
    if u ≠ r then
        A = A ∪ {(p(u), u)}
    for each neighbor v of u do
        if v ∈ Q and w(u, v) < key(v) then
            key(v) ← w(u, v)
            DecreaseKey (key(v), v)
            p(v) ← u
```

Q is a priority queue maintaining distances of vertices not in the tree so far, $\mathsf{key}(v)$ is the minimum weight of edge connecting v to some vertex in the tree, and $p(v)$ is the parent of v in the tree. `ExtractMin` is used to select the node u that is closest to the partial spanning tree but not in it yet according to the value of $\mathsf{key}(u)$, which will be added to the spanning tree at that step. `DecreaseKey` is used to update the weight of nodes (distance to the tree) not yet included in the tree. As a result, a node that was thought to be too far may now move closer to the top of the queue (and therefore would be `ExtractMin`-ed sooner).

(a) How many layers of GNN are needed for a graph with $|V|$ to perform this task? Please give a function related to $|V|$.

(b) For node v, layer k+1, describe a message function $M(h_v^k)$, an aggregation function $h_{N(v)}^{k+1}$, and an update rule h_v^{k+1} for the GNN such that it learns the Prim's Algorithm perfectly. Also, describe a parameter p_v^{k+1} that keeps track of v's predecessor in the constructed spanning tree after updating h_v^{k+1}.

$$M(h_v^k) =$$

$$h_{N(v)}^{k+1} =$$

$$h_v^{k+1} =$$

$$p_v^{k+1} =$$

(Hints:

1. $h_v^k \in \{0,1\}$. $h_v^k = 1$ if node v has been added to the (partial) spanning tree in or before round k. Otherwise $h_v^k = 0$. $h_v^1 = 1$ if and only if $v = r$.

2. You can use argmin/argmax, min/max and parameters that don't occur in the description to describe the required functions. Please specify your notation in your answer.

3. You can use UNDEFINED to describe values that haven't been assigned or initally assigned as NIL.

4. Unlike breadth-first search for reachability, and the Bellman-Ford algorithm for shortest paths, single iterations of Prim's algorithm will specifically focus on one node at a time.

5. Please submit your answer to the four equations above, you may need to discuss different conditions for the same equation.)

14 Computational Advertising (15 points)

In this question, we will consider a new algorithm for computational advertising called k-Greedy-BALANCE. Before we explain the problem, please note that we will restrict the input space to the following constraints:

1. There are three types of queries: x, y, and z.
2. There are two ads: 1. Ad A, which can be shown for queries x and y, and 2. Ad C, which can be shown for queries x and z.
3. Both ads have a budget of $\$B$ and the same bid amount of \$1 for all types of queries.
4. The total number of queries is $2B$.
5. **Only allowed configurations are where the optimal advertising solution achieves $\$2B$ revenue.**

Now, let's describe the k-Greedy-BALANCE algorithm:

1. Read the query.
2. If only one of the ads A or C can be served for the query, serve it.
3. If both ads A and C can be served and have unspent budget:
   - Serve ad C if the unspent budget of C is at least $\$k$ more than the unspent budget of A (i.e., $\text{unspent\_budget}(C) \geq \text{unspent\_budget}(A) + \$k$).
   - Otherwise, serve ad A.

For the following questions, we will assume B is a multiple of 4. Remember that all competitive ratios are calculated using the input space defined under the aforementioned constraints. Also remember from the lectures that within our assumptions, the competitive ratio of the Greedy algorithm is $\frac{1}{2}$, and the competitive ratio of the BALANCE algorithm is $\frac{3}{4}$. You are free to use any content or result proven in the lectures as is, just remember to cite that the result/content has been proven/used in lectures.

Remember that for the purpose of calculating competitive ratio, our input space is restricted to configurations where the optimal advertising solution achieves $2B$ revenue.

1. **(2 point)** Argue that the competitive ratio of B-Greedy-BALANCE is $\frac{1}{2}$.
2. **(2 point)** Argue that the competitive ratio of 1-Greedy-BALANCE is $\frac{3}{4}$.
3. **(5 points)** Show that the competitive ratio of $\frac{B}{4}$-Greedy-BALANCE is greater than $\frac{1}{2}$. (Hint: Think about why the worst-case scenario of the normal Greedy algorithm wouldn't work here.)
4. **(6 points)** Show that the competitive ratio of $\frac{B}{4}$-Greedy-BALANCE within our input constraints is $\frac{11}{16}$.