CS 630, Fall 2024, Homework 7 Solutions
Due Wednesday, November 20, 2024, 11:59 pm EST, via Gradescope

Homework Guidelines

Collaboration policy Collaboration on homework problems, with the exception of programming assignments and reading quizzes, is permitted, but not encouraged. If you choose to collaborate on some problems, you are allowed to discuss each problem with at most 5 other students currently enrolled in the class. Before working with others on a problem, you should think about it yourself for at least 45 minutes. Finding answers to problems on the Web or from other outside sources (including generative AI tools or anyone not enrolled in the class) is strictly forbidden.

You must write up each problem solution by yourself without assistance, even if you collaborate with others to solve the problem. You must also identify your collaborators. If you did not work with anyone, you should write "Collaborators: none." It is a violation of this policy to submit a problem solution that you cannot orally explain to an instructor or TA.

Typesetting Solutions should be typed and submitted as a PDF file on Gradescope. You may use any program you like to type your solutions. LaTeX, or "Latex", is commonly used for technical writing (`overleaf.com` is a free web-based platform for writing in Latex) since it handles math very well. Word, Google Docs, Markdown or other software are also fine.

Solution guidelines For problems that require you to provide an algorithm, you must provide:

1. pseudocode and, if helpful, a precise description of the algorithm in English. As always, pseudocode should include

   - A clear description of the inputs and outputs
   - Any assumptions you are making about the input (format, for example)
   - Instructions that are clear enough that a classmate who hasn't thought about the problem yet would understand how to turn them into working code. Inputs and outputs of any subroutines should be clear, data structures should be explained, etc.

   If the algorithm is not clear enough for graders to understand easily, it may not be graded.

2. a proof of correctness,

3. an analysis of running time and space.

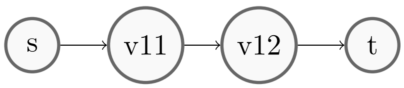
You may use algorithms from class as subroutines. You may also use facts that we proved in class.

You should be as clear and concise as possible in your write-up of solutions. A simple, direct analysis is worth more points than a convoluted one, both because it is simpler and less prone to error and because it is easier to read and understand.

Problem 1 *Multi-path routing (10 points)*

The following scenarios consider multi-path routing in an unreliable network where transmissions are only successful with probability p. The goal in each scenario is to get a message from s to t. Each link (edge) transmits at most one time, and you can assume that the sending node waits until all successful incoming messages have been received, and that each transmission is independent of the others.

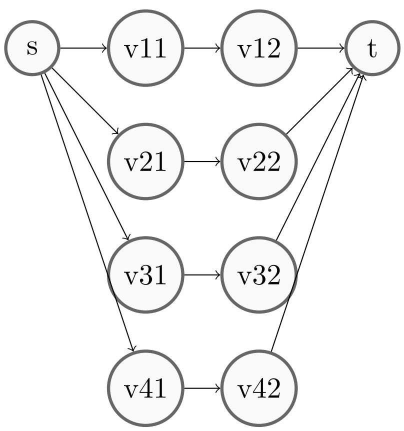
1. Consider the following graph with 1 path from s to t. Write the probability that a message from s is received by t as a function of p, and evaluate that probability for $p = 0.5$.



Solution. There are three transmissions along the path, and each transmission succeeds independently with probability p, so the probability of success is p^3.

For $p = 0.5$, that probability is 0.125.

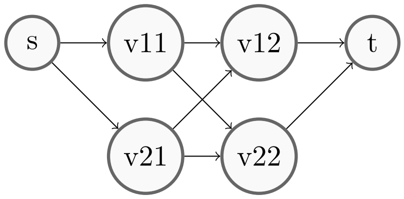
2. Consider the following graph with 4 non-overlapping paths from s to t. Write the probability that a message from s is received by t as a function of p, and evaluate that probability for $p = 0.5$ to the nearest hundredth.



Solution. The four paths do not share any edges, so each path succeeds independently of the others. The probability that one path fails is $1 - p^3$, so the probability that all paths fail is $(1 - p^3)^4$, and the probability that at least one path succeeds is $1 - (1 - p^3)^4$.

For $p = 0.5$, that probability is approximately 0.41.

3. Now consider the following graph with 4 overlapping paths from s to t. What is the probability that a message from s is received by t when $p = 0.5$? You may round your final answer to the nearest hundredth.



Solution. To simplify this analysis, we will consider the graph in layers in the following groups – $v11$ and $v21$, $v12$ and $v22$, and finally t. Because of symmetry in the graph, we only care how many of nodes in the group receive the message.

The following table shows the probabilities of different combinations of $v11$ and $v21$ receiving the message.

Neither $v11$ nor $v12$	Exactly one of $v11$ and $v12$	Both $v11$ and $v21$
$(1-p)^2$	$2p(1-p)$	p^2

The following table shows the conditional probabilities of $v12$ and $v22$ receiving the message conditioned on how many of $v11$ and $v21$ received the message.

	Neither $v12$ nor $v22$	Exactly one of $v12$ and $v22$	Both $v12$ and $v22$
Neither $v11$ nor $v12$	1	0	0
Exactly one of $v11$ and $v12$	$(1-p)^2$	$2p(1-p)$	p^2
Both $v11$ and $v21$	$(1-p)^4$	$2(1-(1-p)^2)(1-p)^2$	$(1-(1-p)^2)^2$

The following table shows the conditional probabilities of t receiving the message.

	Not t	t
Neither $v12$ nor $v22$	1	0
Exactly one of $v12$ and $v22$	1-p	p
Both $v12$ and $v22$	$(1-p)^2$	$1-(1-p)^2$

Matrix multiplication gives us the end-to-end probabilities. Plugging in $p = 0.5$, we get the following calculation.

$$\begin{bmatrix} 0.25 & 0.5 & 0.25 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0.25 & 0.5 & 0.25 \\ 0.0625 & 0.375 & 0.5625 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0.5 & 0.5 \\ 0.25 & 0.75 \end{bmatrix} = \begin{bmatrix} 0.62890625 & 0.37109375 \end{bmatrix}$$

So when $p = 0.5$, the probability that node t receives the message is 0.37109375.

Problem 2 *MAX Sketches (10 points)*

Often, instead of storing every element of some data we only keep some aggregate information on it. We can design many different data structures for this depending on our application. We refer to these clever data structure designs as sketches.

Here is the definition of one such sketch. Answer the questions below about this design.

- Each sketch S is an unsigned 32 bit integer with values from 0 to $2^{32} - 1$.
- Initially S is set to 0.
- An item x is inserted into S by the following procedure; first we calculate the hash $h(x)$, which is an integer in the same range 0 to $2^{32} - 1$. Then we update S to be $S = \max(S, h(x))$.

As an example: Suppose the items x, y and z are inserted one at a time to S. They have hash values $h(x) = 3, h(y) = 10$ and $h(z) = 7$. Initially $S = 0$. First we add x and get $S = \max(0, 3) = 3$. Next for y we get $S = max(3, 10) = 10$. Finally when inserting z, S won't change $S = \max(10, 7) = 10$.

1. Given a sketch S_A which had elements of set A inserted, and sketch S_B which had elements of set B inserted, explain how to calculate $S_{A\cup B}$ which had the elements of both A and B inserted.

   Solution. By construction, $S_X = \max(0, \max(h(x)|x \in X))$ aside from the case where $X = \emptyset$. So

$$\begin{aligned} S_{A\cup B} &= \max(0, \max(h(x)|x \in A \cup B)) \\ &= \max(0, \max(h(x)|x \in A), \max(h(x)|x \in B)) \\ &= \max(S_A, S_B) \end{aligned}$$

2. Given a sketch S which had one item x inserted, what is the probability that $S \geq 2^{31}$?

   Solution. $h(x)$ is equally likely to fall in $[0, 2^{31} - 1]$ and $[2^{31}, 2^{32} - 1]$, so the probability is 1/2.

3. Given a sketch S which had one item inserted and integer i between 1 and 31, what is the probability that $S \geq 2^{32} - 2^{32-i}$?

   Solution. There are 2^{32-i} possible hash values in the range $[2^{32} - 2^{32-i}, 2^{32} - 1]$, so the probability is $\frac{2^{32-i}}{2^{32}} = 1/2^i$.

4. Given a sketch S with 2^k items inserted with $k \leq 25$, give a lower bound on the value for S which holds with probability at least 1/2? For full credit, your lower bound must be within a factor of 4 of the best possible lower bound as measured from the maximum hash $2^{32} - 1$, so the trivial lower bound 0 does not qualify.

   Solution. After 2^k insertions, the probability that at least one hash is at least $2^{32} - 2^{32-k}$ is $1 - (1 - 1/2^k)^{2^k} \approx 1 - 1/e > 0.5$.

   Is that tight enough? The probability that at least one hash is at least $2^{32} - 2^{32-k-1}$ is $1 - (1 - 1/2^{k+1})^{2^k} \approx 1 - 1/e^{1/2} < 0.4$ so it is within a factor of two from optimal.