1. (10 points) **From Independent Set to Coloring.** Given a graph $G = (V, E)$ with n vertices and maximum degree Δ which is 3-colourable, in class we saw a method which can round an SDP and recover an independent set of size at least $c\frac{n}{\Delta^{1/3}}$, for some constant c (actually we had some logarithmic factors inside, but let us ignore that for simplicity here; also, our algorithm was randomized, but ignore that as well for simplicity and assume it always finds an independent set of this size). Using this as a sub-routine, give a procedure to colour the graph using as few colours. How many colours do you use?

Solution: We give a two step algorithm that proceeds as follows

1. While maximum degree of graph satisfies $\Delta \geq n^{3/4}$, we keep doing the following
   - Give a new colour c_1 to the maximum degree vertex.
   - The neighbours of the maximum degree vertex can be coloured with at most 2 new colours c_2, c_3. This is true since the original graph is 3-colourable.
   - Remove the coloured $\Delta + 1$ vertices from the current graph.
2. Now, we have current maximum degree $\Delta < n^{3/4}$. Thus, using the subroutine, the size of independent set is at least

$$c\frac{n}{\Delta^{1/3}} \geq c\frac{n}{n^{1/4}} = cn^{3/4}$$

   Each independent set can be given one new colour and the corresponding vertices are removed iteratively.

In each iteration of step 1, we remove at least $n^{3/4}$ vertices and introduce at most 3 colours. So, maximum number of colours introduced in step 1 is bounded by

$$\frac{n}{n^{3/4}} * 3 = 3.n^{1/4} = O(n^{1/4})$$

In each iteration of step 2, we use one colour and the number of vertices removed is at least $c\frac{n}{n^{1/4}}$ (where n is the remaining number of vertices at that iteration).

We now show that when you iteratively keep removing $c\frac{n}{n^p}$ from n (where n is currently remaining elements, $c > 0$ and $0 < p < 1$), then the total number of iterations till remaining elements goes down to 1 is bounded by $\frac{1}{cp}n^p$. We prove this by considering the limiting case of an integral. Here we have y as the remaining number of elements and x as the number of time steps.

$$\frac{dy}{dx} = -c\frac{y}{y^p} = -cy^{1-p}$$

$$\implies -\int_n^0 y^{p-1}dy = \int_0^X c dx$$

$$\implies \left[\frac{y^p}{p}\right]_0^n = cX$$

$$\implies \#\ of\ steps\ = X \approx \frac{1}{cp}n^p$$

Using this in step 2 ($p = \frac{1}{4}$), we have total new colours used in step 2 as at most

$$\frac{4}{c}n^{1/4}.1 = O(n^{1/4})$$

Since both steps use $O(n^{1/4})$ new colours, overall number of colours used by the algorithm is also $O(n^{1/4})$.

2. (70 points) **(How long to lease, before you buy)** In this problem, we explore the primal-dual method for designing online algorithms. In the problem set-up, we just moved into a city and plan to stay there till the end of a project we are working on. The problem is, we don't know how long the project will last. So when you're getting a modem for the internet at home, the company gives us two choices: (a) rent the modem at 1 rupee per day, or (b) buy it for a fixed cost of B rupees.

   (a) (2 points) If we knew the number of days we're going to stay, what is the optimal thing to do to minimize our cost?

   Solution: Let D be the number of days we'll stay. If we know we will end up buying the modem, there is no need to rent it initially. So,

   - if $B \leq D$, buy modem
   - else, rent modem

   (b) (5 points) Since we don't know the number of days until the project is over, can we devise a good algorithm with small competitive ratio, i.e., ratio of our cost to the optimal cost for the given input? For example, one such policy will be to lease for a certain number of days, and then buy the modem. What's the best choice of how many days to lease, to achieve a competitive ratio of 2.

   Solution: The competitive ratio is the maximum (worst case over all problem instances) ratio of our algorithm's solution to the optimal solution. Consider the following algorithm:- On Day i,

   - if $1 \leq i \leq B - 1$, rent modem for that day

- on Day $i = B$, buy the modem and use it till project ends.

We now analyze the ratio

- If $1 \le D \le B - 1$, both our solution and optimal solution are the same, to rent for D days. So ratio is 1
- If $D = B + x$ for some $x \ge 0$, optimal solution is to buy on Day 1 ($cost = B$). Our solution is to rent for $B - 1$ days and buy on B^{th} day ($cost = B - 1 + B$).

$$\implies competitive\ ratio = \max\left(1, \frac{2B-1}{B}\right) \le 2$$

(c) (5 points) Show that no deterministic algorithm can do better than 2-competitive.

Solution: Any reasonable deterministic algorithm will rent for the first few days (say x days, $x > 0$) and buy on the $(x+1)^{th}$ day. (Renting after buying is unnecessary). We consider a worst case problem instance I that has $D = x + 1$. Thus $COST(Algo(I)) = x + B$. But, from question 2.a, we have $COST(OPT(I)) = min(B, D) = min(B, x + 1)$.

- If $x \ge B$

$$\frac{COST(Algo(I))}{COST(OPT(I))} = \frac{x + B}{min(B, x+1)} \ge \frac{B + B}{B} = 2$$

- If $x \le B - 1 \implies B \ge x + 1$

$$\frac{COST(Algo(I))}{COST(OPT(I))} = \frac{x + B}{min(B, x+1)} \ge \frac{x + x + 1}{x + 1} = 2\ (in\ the\ limit)$$

Thus no deterministic algorithm can do better than 2-competitive (in the limit).

(d) (10 points) Now we show that we can do better with randomized algorithms. For this, consider the following algorithm: keep leasing for the first $3B/4$ days. Then with probability 0.5, buy the modem on day $3B/4$. With the remaining probability, continue leasing until day B. At this point, if we haven't already bought it, buy the modem. Show that the competitive ratio (which is maximum over all T of the expected cost of the algorithm if we lived in the city for T days divided by optimal cost for the same T days) is better than 2.

Solution:

- if $D < 3B/4$,
  then both our algorithm and optimal solution only rents. $COST(Algo(I)) = COST(OPT(I)) = D$. Thus ratio is 1.

- if $3B/4 \le D < B$,
  then $COST(OPT(I)) = D$. If our algo buys on day $3B/4$, then $COST(Algo(I)) = (3B/4 - 1) + B \approx 7B/4$. Else, $COST(Algo(I)) = D$.

$$\implies \mathbb{E}[COST(Algo(I))] = 0.5 * 7B/4 + 0.5 * D = 7B/8 + D/2$$

  Since $3B/4 \le D \implies \frac{B}{D} \le \frac{4}{3}$

$$\implies \frac{\mathbb{E}[COST(Algo(I))]}{COST(OPT(I))} = \frac{7B/8 + D/2}{D} = \frac{1}{2} + \frac{7}{8}.\frac{B}{D} \le \frac{1}{2} + \frac{7}{8}.\frac{4}{3} = \frac{5}{3}$$

- if $B \le D$,
  then $COST(OPT(I)) = B$. If our algo buys on day $3B/4$, then $COST(Algo(I)) = (3B/4 - 1) + B \approx 7B/4$. Else, it buys on day B and $COST(Algo(I)) = (B - 1) + B \approx 2B$.

$$\implies \mathbb{E}[COST(Algo(I))] = 0.5 * 7B/4 + 0.5 * 2B = 15B/8$$

$$\implies \frac{\mathbb{E}[COST(Algo(I))]}{COST(OPT(I))} = \frac{15B/8}{B} = \frac{15}{8}$$

$$\implies \textit{competitive ratio} = \max(1,\ 5/3,\ 15/8) = 15/8\ \ (\textit{better than } 2)$$

(e) (3 points) We now devise a primal-dual based online algorithm to determine the optimal probabilities of when to buy and when to lease. To this end, consider the following LP relaxation of the problem. $\min(\sum_{1 \le t \le T} x_t) + B \cdot y$ subject to $x_t + y \ge 1$ for all $1 \le t \le T$ and $x_t, y \ge 0$. Show that this is indeed a relaxation of the optimal solution if we lived in the city for T days.

Solution: The variable x_t indicates if the optimal solution rented the modem on day t (1 if true). The variable y indicates if the optimal solution purchased the modem at any point. From the optimal solution described in question 2.a,

- if $T < B$, then the optimal solution rents the modem on all days. Thus $x_t = 1,\ \forall t = 1\ \textit{to}\ T$ and $y = 0$. This satisfies the constraints of the LP. Also, the objective function $\sum_{1 \le t \le T} x_t + B \cdot y = \sum_{1 \le t \le T} 1 + B \cdot 0 = T$ has the same value as that of the solution's cost.

- if $T \geq B$, then the optimal solution buys the modem on the first day. Thus $x_t = 0,\ \forall t = 1\ to\ T$ and $y = 1$. This satisfies the constraints of the LP. Also, the objective function $\sum_{1 \leq t \leq T} x_t + B \cdot y = \sum_{0 \leq t \leq T} 1 + B \cdot 1 = B$ has the same value as that of the solution's cost.

Since any optimal solution to the problem is a feasible solution to the LP, the LP is a relaxation of the optimal solution.

(f) (5 points) Construct the dual of this LP using variables z_t. Write down the objective function and constraints.

Solution: The t^{th} primal constraint is multiplied by z_t to get the following dual LP.

$$\max \sum_{t=1}^{T} z_t$$

such that

$$z_t \leq 1\ \ \forall\, t = 1\ to\ T$$

.

$$\sum_{t=1}^{T} z_t \leq B$$

$$z_t \geq 0\ \ \forall\, t = 1\ to\ T$$

(g) (10 points) Now in the online problem, the constraints $x_t + y \geq 1$ appear one by one, i.e., for each new day we are living in the city, the new constraint appears we need to satisfy. So we first construct a fractional solution with small cost as follows. Initialize $y \leftarrow 0$. On each new day t, if $y < 1$, then update $y \leftarrow y(1 + \frac{1}{B}) + \frac{1}{cB}$ and, set $x_t = 1 - y$. Correspondingly, we also update $z_t \leftarrow 1$ for the dual. On the other hand, if $y \geq 1$ then do nothing both to primal and dual. We will soon determine the value of c. For this update rule, first show that the primal solution we maintain is a feasible fractional solution. Moreover, show that the increase in primal objective is $1 + 1/c$ while the increase in dual objective is 1.

Solution: Let y_t denote the value of y obtained on day t ($y_0 = 0$). We show by induction that the primal solution we maintain is a feasible fractional solution:-

- One Day $t = 1$, we have $y_1 = 0 + \frac{1}{cB} = \frac{1}{cB}$ and $x_1 = 1 - 0 = 0$. Thus it satisfies $x_t + y_1 \geq 1$ for first day.

- Assume $x_t + y_d \geq 1$ is satisfied for all days 1 to d seen so far.

- On the $(d+1)^{th}$ day, we have 2 cases and we show that condition is satisfied in both cases
  - if $y_d \geq 1$, then $y_{d+1} = y_d \geq 1$. Hence, since $x_t \geq 0$, it satisfies $x_t + y_{d+1} \geq 1$ for all t from 1 to $d+1$.
  - if $y_d < 1$, then $y_{d+1} = y_d(1 + \frac{1}{B}) + \frac{1}{cB} \geq y_d$ and $x_{d+1} = 1 - y_d$. Hence, $x_{d+1} + y_{d+1} \geq x_{d+1} + y_d = 1$. Also, for previous days with $1 \leq t \leq d$, we have $x_t + y_{d+1} \geq x_t + y_d \geq 1$ (by assumption). Thus the condition is still satisfied for all days t from 1 to $d+1$.
- Hence by Principle of Mathematical Induction, the primal solution we maintain is always a feasible fractional solution.

Now, let the value of primal objective at the end of day t be P_t. We now consider the change to primal objective after a day when y is updated (i.e, case $y_d < 1$).

$$\begin{aligned}
P_{d+1} - P_d &= \left(\sum_{1\leq t\leq d+1} x_t + B \cdot y_{d+1}\right) - \left(\sum_{1\leq t\leq d} x_t + B \cdot y_d\right) \\
&= x_{d+1} + B(y_{d+1} - y_d) \\
&= (1 - y_d) + B\left(\left(y_d\left(1 + \frac{1}{B}\right) + \frac{1}{cB}\right) - y_d\right) \\
&= 1 + \frac{1}{c}
\end{aligned}$$

Now, let the value of dual objective at the end of day t be Q_t. We now consider the change to dual objective after a day when y is updated (i.e, case $y_d < 1$. this implies $z_{d+1} = 1$).

$$Q_{d+1} - Q_d = \sum_{t=1}^{d+1} z_t - \sum_{t=1}^{d+} z_t = z_{d+1} = 1$$

(h) (10 points) Now, for what value of c will the dual be feasible? So note that whenever we set $z_t \leftarrow 1$, the y value increases. So set c so that the y variable reaches 1 by time $t = B$ and show that this will imply that the dual we construct is a feasible dual solution.

Solution: Let y_t denote the value of y obtained on day t ($y_0 = 0$). If we assume that y is updated on every day seen so far, and we expand the expression for y_t using the update expression, we get the formula that

$$y_t = \frac{1}{c}\left[\left(1 + \frac{1}{B}\right)^t - 1\right]$$

This expression can be proved by induction by the facts that

- Day 1 has $y_1 = \frac{1}{cB} = \frac{1}{c}\left[\left(1+\frac{1}{B}\right)^1 - 1\right]$
- Assuming the expression is true for day t, we see that it is also true for day $t+1$.

$$\begin{aligned} y_{t+1} &= y_t\left(1+\frac{1}{B}\right) + \frac{1}{cB} \\ &= \frac{1}{c}\left[\left(1+\frac{1}{B}\right)^t - 1\right]\left(1+\frac{1}{B}\right) + \frac{1}{cB} \\ &= \frac{1}{c}\left[\left(1+\frac{1}{B}\right)^{t+1} - 1\right] \end{aligned}$$

We want y variable to reach 1 by the time $t = B$. So, we want $y_B = 1$. Putting $t = B$ in the expression gives

$$1 = y_B = \frac{1}{c}\left[\left(1+\frac{1}{B}\right)^B - 1\right]$$

$$\implies c = \left(1+\frac{1}{B}\right)^B - 1$$

Thus, in the limit of a large B, we have

$$\implies c = \lim_{B\to\infty}\left(1+\frac{1}{B}\right)^B - 1 = e - 1$$

For $c = e-1$, it can be seen that $y_t = \frac{1}{e-1}\left[\left(1+\frac{1}{B}\right)^t - 1\right]$ always satisfies $y_B \leq 1$ and $y_{B+1} > 1$ (equality only in limit). Hence, based on the update rules of the algorithm, y will be updated exactly on the first B days. Thus correspondingly, we have $z_t = 1$ only for the first B days and $z_t = 0$ for the remaining days. Thus the dual constraint $\sum_{t=1}^{T} z_t \leq B$ is satisfied and it is a feasible dual solution.

(i) (10 points) Using the above parts, show that our fractional cost has competitive ratio of roughly $e/(e-1)$.

Solution: For $c = e-1$, we have seen in section 2.h that y is updated (i.e, $y < 1$) only till the B^{th} day. Also, we have seen in section 2.g that whenever y is updated, the primal objective is increased by $1 + \frac{1}{c} = 1 + \frac{1}{e-1} = \frac{e}{e-1}$. Primal objective doesn't increase when y is not updated. Thus, assuming true number of days of project to be D, we have

- If $1 \leq D \leq B - 1$,
  then our fractional solution updates y on all of these days. Optimal solution rents on all days.

$$\frac{COST(Algo(I))}{COST(OPT(I))} = \frac{D.\frac{e}{e-1}}{D} = \frac{e}{e-1}$$

- If $D \geq B$, then optimal solution buys on first day. Fractional solution updates y on the first B days (till $y \geq 1$) and will not update on remaining days.

$$\frac{COST(Algo(I))}{COST(OPT(I))} = \frac{B.\frac{e}{e-1}}{B} = \frac{e}{e-1}$$

$$\implies competitive\ ratio = \max\left(\frac{e}{e-1}, \frac{e}{e-1}\right) = \frac{e}{e-1}$$

(j) (10 points) Finally, for the rounding. Suppose we choose a random real value τ uniformly between 0 and 1. Then we buy the modem on the first day the y value exceeds τ (until then we keep leasing). Show that the competitive ratio of this algorithm is also the same as the fractional competitive ratio identified above. This shows the power of randomization, and also the use of the primal-dual method for online algorithms!

Solution: Once again, let y_t denote the value of y obtained on day t ($y_0 = 0$). From the update rule $y \leftarrow y(1 + \frac{1}{B}) + \frac{1}{cB}$, we have the result

$$B.(y_{t+1} - y_t) = y_t + \frac{1}{c} \tag{1}$$

We use this to compute the expected cost of the online randomized algorithm (with $c = e-1$). Let the true number of days of the project be D and let $D < B$ for now. Then, for a uniformly chosen τ between 0 and 1,

- Algorithm buys on day 1 if $y_0 \leq \tau < y_1$ (probability $= y_1 - y_0$)
- Algorithm buys on day 2 if $y_1 \leq \tau < y_2$ (probability $= y_2 - y_1$), and so on till
- Algorithm buys on day D if $y_{D-1} \leq \tau < y_D$ (probability $= y_D - y_{D-1}$)
- Algorithm doesn't buy till the end if $y_D \leq \tau < 1$ (probability $= 1 - y_D$)

Thus, using these probabilities and the corresponding values of cost, expected cost of algorithm is given by

$$
\begin{aligned}
&\mathbb{E}[COST(ALGO(I))] \\
&= \left[\sum_{i=0}^{D-1}(y_{i+1}-y_i)(B+i)\right] + (1-y_D)D \\
&= \left[\sum_{i=0}^{D-1}(y_{i+1}-y_i)i\right] + \left[\sum_{i=0}^{D-1}(y_{i+1}-y_i)B\right] + (1-y_D)D
\end{aligned}
$$

Using telescopic summation and equation 1, we have

$$
\begin{aligned}
&= \left[\sum_{i=1}^{D}-y_i\right] + \left[\sum_{i=0}^{D-1}(y_i+\frac{1}{c})\right] + D \\
&= \left[\sum_{i=1}^{D}-y_i\right] + \left[\sum_{i=0}^{D-1}y_i\right] + D.\frac{1}{c} + D \\
&= D\left(1+\frac{1}{c}\right) - y_D \qquad (as\ y_0 = 0)
\end{aligned}
$$

Since y_D is a fraction at most 1, we have

$$
\frac{\mathbb{E}[COST(ALGO(I))]}{COST(OPT(I))} = \frac{D\left(1+\frac{1}{c}\right)-y_D}{D} \leq \frac{D\left(1+\frac{1}{c}\right)}{D} = 1+\frac{1}{c} = \frac{e}{e-1}
$$

The above is an equality in the limit when D is large. Now, consider the case when $D \geq B$. Optimal solution has cost B. Since our algorithm always buys modem on or before day B, it has the same cost for all $D \geq B$. Further, we can see that the algorithm's cost for $D = B$ can be obtained by using $D = B$ in our previous derivation of expected cost (since $y_D = 1$). Thus,

$$
\frac{\mathbb{E}[COST(ALGO(I))]}{COST(OPT(I))} = \frac{B\left(1+\frac{1}{c}\right)-y_B}{B} \leq \frac{B\left(1+\frac{1}{c}\right)}{B} = 1+\frac{1}{c} = \frac{e}{e-1}
$$

Thus the randomized algorithm has the same required competitive ratio in both cases.

3. (20 points) **Secure Algorithms.** We live in a world where we need to constantly guard ourselves, and our data from hack attacks and privacy breaches. In this problem, imagine you are a statistics company and need to maintain some aggregate information of some individual records, and should build our system in such a way that even if our database (or index we store in memory) is leaked, it reveals as little as possible about the individual records. As a toy example, suppose each day, we receive information about a

new vector $v_t \in \mathbb{R}^n$. We would like to always maintain a good estimate (approximation is allowed), of the ℓ_2-norm of the sum of the vectors $\sum_{t' \leq t} v_t$. Using Gaussian ideas we learnt in class, can we preserve a good estimate of $||\sum_{t' \leq t} v_t||$? We would like to build an index of memory usage $O(n)$, but also the contents of the memory must reveal as little about the individual component vectors as possible, perhaps nothing beyond what we can anyway learn from the estimate value which the algorithm must maintain at all times. Can we design such a scheme?

Solution: We make use of the 2-stable property of the Gaussian distribution. Let $v \in \mathbb{R}^n$ be a given vector and let $w_1, w_2, \ldots, w_n$ be iid sampled from $\mathcal{N}(0, 1)$. Then, we saw in class that $\langle v, w \rangle$ is equivalent to a Gaussian RV sampled from $\mathcal{N}(0, ||v||^2)$. Further, for any Gaussian RV $X \leftarrow \mathcal{N}(0, \sigma^2)$, the corresponding half normal RV $Y = |X|$ has

$$\mathbb{E}[Y] = \mathbb{E}[|X|] = \int_0^\infty 2.\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}.xdx = \sigma\sqrt{\frac{2}{\pi}} \int_\infty^0 \frac{-x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}} dx = \sigma\sqrt{\frac{2}{\pi}} \left[e^{-\frac{x^2}{2\sigma^2}} \right]_\infty^0 = \sigma\sqrt{\frac{2}{\pi}}$$

Similarly, the variance of Y is gotten as

$$var(Y) = \mathbb{E}[Y^2] - (\mathbb{E}[Y])^2 = \mathbb{E}[|X|^2] - \left(\sigma\sqrt{\frac{2}{\pi}}\right)^2 = \mathbb{E}[X^2] - \frac{2\sigma^2}{\pi} = \sigma^2 - \frac{2\sigma^2}{\pi} = \sigma^2\left(1 - \frac{2}{\pi}\right)$$

Now consider the following secure scheme

- For a chosen constant c, we maintain c Gaussian vectors of n dimensions, where each component of each vector is a iid sample from $\mathcal{N}(0, 1)$.
- For each Gaussian vector g_i, we maintain a corresponding value x_i initialized to 0.
- Whenever a new vector v_t is added, we update all x_i values as $x_i = x_i + \langle v_t, g_i \rangle$.
- Whenever a query for $||\sum_{t' \leq t} v_t||$ is made, we return $y = \frac{1}{c}\sqrt{\frac{\pi}{2}}(\sum_i |x_i|)$

Memory complexity of algorithm is $nc + c = O(n)$.
Let $s_t = \sum_{t' \leq t} v_t$ be the sum at time t. Since the g_i vectors are spherically random and we only additionally store the dot products $\langle s_t, g_i \rangle$, the adversary does not get any additional info about the individual vectors v_t. We now show why y is a good estimate for $||s_t||$.
$x_i = \langle s_t, g_i \rangle$. Thus $x_i \leftarrow \mathcal{N}(0, ||s_t||^2)$. So, using the above seen properties of half normal RVs.

$$\mathbb{E}[|x_i|] = \sigma\sqrt{\frac{2}{\pi}} = ||s_t||\sqrt{\frac{2}{\pi}}$$

$$\mathbb{E}[y] = \frac{1}{c}\sqrt{\frac{\pi}{2}}\left(\mathbb{E}\left[\sum_i |x_i|\right]\right) = \frac{1}{c}\sqrt{\frac{\pi}{2}}\left(\sum_i \mathbb{E}[|x_i|]\right) = \frac{1}{c}\sqrt{\frac{\pi}{2}}\left(\sum_i ||s_t||\sqrt{\frac{2}{\pi}}\right)$$

$$\implies \mu_y = \mathbb{E}[y] = \frac{1}{c}\sqrt{\frac{\pi}{2}}\left(c||s_t||\sqrt{\frac{2}{\pi}}\right) = ||s_t||$$

Thus the expected value of y is the required norm. Further, since each x_i (and hence $|x_i|$) is independent of each other, we use the variance of half normal RVs to get

$$var(y) = \left(\frac{1}{c}\sqrt{\frac{\pi}{2}}\right)^2 var\left(\sum_i |x_i|\right) = \frac{\pi}{2c^2}\sum_i var\left(|x_i|\right) = \frac{\pi}{2c^2}\sum_i ||s_t||^2\left(1-\frac{2}{\pi}\right)$$

$$\implies var(y) = \frac{\pi}{2c^2}.c||s_t||^2\left(1-\frac{2}{\pi}\right) = \frac{\pi-2}{2c}||s_t||^2$$

$$\implies \sigma_y = \sqrt{var(y)} = \sqrt{\frac{\pi-2}{2c}}||s_t||$$

Now, Chebyshev's inequality gives that for any $t \geq 0$, we have

$$Pr\left(|y-\mu_y| \geq t.\sigma_y\right) \leq \frac{1}{t^2}$$

$$\implies Pr\left(\frac{|y-||s_t|||}{||s_t||} \geq t.\sqrt{\frac{\pi-2}{2c}}\right) \leq \frac{1}{t^2}$$

$$\implies Pr\left(relative\ error\ in\ computed\ norm \geq t.\sqrt{\frac{\pi-2}{2c}}\right) \leq \frac{1}{t^2}$$

Thus, we can increase c to lower relative error of approximation as required.

4. (0 points) **Difficulty Level.** How difficult was this homework? How much time would you have spent on these questions?

Solution: Roughly around 12 hours.