ID: 03 **Collaborator #1:** Last Name, First Name
Name: Al-Naji, Nader **Collaborator #2:** Last Name, First Name

Solution to HW 5, Problem 1

COS 340 - Spring 2012

There are b boys and g girls arranged in a line. Let $n = b+g$ be the number of students. Let s be the number of locations where a boy and a girl stand next to each other. For example, for BBGGBBGGGB, the value of s is 4 and n is 10.

1. Count the number of arrangements in which s is even.

Call a place where a boy and a girl stand next to each other in the line a "shift." Using this language, it should be clear that a line with an even number of shifts must start and end with a person of the same gender. To convince yourself of this, simply consider that if the number of shifts is even, then, starting from the first person in line, every shift to a person of a gender different from that of the gender of the first person in line must be accompanied by a shift back to a person of the same gender. So every shift to a different gender must be paired with a shift back in order for the number of shifts to be even. It should also be stated that, by the same logic, that if the gender of the first person in line is the same as the gender of the last person in line, then the number of shifts is even. Finally, note that every arrangement in which s is even maps to one and only one arrangement in which the end-persons have the same gender and vice versa. That is, it is not the case that an arrangement in which s happens to be even counts as two separate arrangements in which the gender of the end-persons is the same and vice versa. Having noted this, we can compute the number of arrangements in which s is even by computing the number of arrangements in which the gender of the end-persons is the same, since the size of both of these sets must be the same (since they form a bijection).

The number of arrangements in which the end-persons have the same gender can be computed by first fixing the gender of the end-persons to be B and permuting the inside $n-2$ persons and then fixing the gender of the end-persons to be G and then permuting the inside $n-2$ persons, and summing these two quantities. Assuming all persons of the same gender are indistinguishable, we thus have that the number of arrangements is:

$\binom{n-2}{b} + \binom{n-2}{g}$.

However, because persons of the same gender are actually distinct, every arrangement must count for $b!g!$ different arrangements since every permutation of the boys and girls while preserving gender location must be counted. Thus, because arrangements in which the boys/girls are distinct form a $k - to - 1$ mapping onto arrangements where the boys/girls are indistinguishable, where $k = b!g!$, we must multiply the number above by $b!g!$ and get:

$b!g![\binom{n-2}{b} + \binom{n-2}{g}]$

Finally, because this only holds for $n > 2$, we have to deal with the degenerate cases where $n = 2$, $n = 1$, and $n = 0$. As the final answer we have, covering all these cases:

$$s(b, g, n = b+g) = \begin{cases} b!g![\binom{n-2}{b} + \binom{n-2}{g}], & \text{if } n > 2 \\ 2, & \text{if } n = 2 \text{ and } b \neq g \\ 0, & \text{if } n = 2 \text{ and } b = g \\ 1, & \text{if } n = 1 \text{ or } n = 0 \end{cases}$$

2. What is the expected value of s if the students are permuted randomly?

Consider the gaps between each boy and girl in the line. Define G_i to be an indicator random variable that is 1 if gap i is a shift and 0 if gap i is not a shift. There are $n-1$ such gaps in any given line of boys and girls and, therefore $n-1$ such random variables. We want the expected number of shifts or, in other words, the expected number of gaps that are 1. But this is simply, using linearity of expectation:

$E[s] = E[\sum_{i=1}^{n-1} G_i] = \sum_{i=1}^{n-1} E[G_i] = \sum_{i=1}^{n-1} E[G_1] = (n-1)G_1$ since the G_i are identically distributed.

It should be clear that the G_i are identically distributed since it doesn't make sense that one gap would be more likely to contain a shift than any other.

So if we can figure out the distribution for G_1, we have our solution. To get the probability that G_i is a shift, we simply sum the probabilities of two events: the event that a girl is on the left and a boy is on the right, and the event that a boy is on the left and a girl is on the right. It should be clear that these are the two cases under which gap i is a shift. Computing this, we get:

$P(\text{space } i \text{ is a shift}) = P(\text{boy on left, girl on right}) + P(\text{girl on left, boy on right})$
$= P(\text{boy on left})P(\text{girl on left} \mid \text{boy on right}) + P(\text{girl on left})P(\text{boy on right} \mid \text{girl on left})$ by the chain rule
$= \frac{b}{n}\frac{g}{n-1} + \frac{g}{n}\frac{b}{n-1} = 2\frac{bg}{n(n-1)}$.

Now that we have the probability that G_i is a shift, we can finish our computation:

$E[s] = (n-1)G_1 = (n-1)2\frac{bg}{n(n-1)} = 2\frac{bg}{n}$.

ID: 03 **Collaborator #1:** Last Name, First Name
Name: Al-Naji, Nader **Collaborator #2:** Last Name, First Name

Solution to HW 5, Problem 2

COS 340 - Spring 2012

A box has n balls each with a different color. We repeatedly draw a random ball from the box and return it. Let R be the random variable that denotes the number of draws until the first time that some ball is picked twice.

1. Compute $P[R > k]$ for any integer $k \geq 0$.

$P[R > k] = \prod_{k=0}^{k-1}(n-k)/n$. We will reduce this to a closed form at the end but first let us prove that this is true by induction.

$Q(k) =$ "$P[R > k] = \prod_{k=0}^{k-1}(n-k)/n$ for all $k \leq n$."

Base: $k = 1$

For $k = 1$, we need to get past the first draw without picking a ball we've seen before. This happens with probability 1 since we haven't seen any balls before, which agrees with the claim.

Step: Assume true for k. Prove true for $k + 1$.

If we want $R > k + 1$, then we need to get through k choosings without picking a ball we've seen before, and then we need to pick a ball we haven't seen before one more time. So $P[R > k+1] = P[R > k] \cdot (n-k)/n = \prod_{k=0}^{(k+1)-1}(n-k)/n$, which agrees with the claim.

Now that it is clear that $P[R > k] = \prod_{k=0}^{k-1}(n-k)/n$, let us reduce this to a simpler expression.

$P[R > k] = \prod_{k=0}^{k-1}(n-k)/n = \frac{(n)(n-1)\ldots(n-k+1)}{n^k} = \frac{n!}{(n-k)!n^k}$
$= \binom{n}{k}\frac{k!}{n^k}$

Finally, note that after n draws, it is impossible to draw a ball we haven't seen before and $P[R > k] = 0$. So, as the final answer we have:

$$P[R > k] = \begin{cases} \binom{n}{k}\frac{k!}{n^k}, & \text{if } 0 \leq k \leq n \\ 0, & \text{otherwise} \end{cases}$$

2. Determine $E[R^2]$ within constant factors. Show your calculations.

We will establish an upper bound and then establish a lower bound. We will start with the upper bound.

First, we need to derive an expression for $E[R^2]$ in terms of $P[R > k]$. To do this, let's take a look at what $E[R^2]$ actually is by definition:

$E[R^2] = \sum_{k=0}^{\infty} P[R = k]k^2 = \sum_{k=0}^{n} P[R = k]k^2$ since $P[R = k] = 0$ for $k > n$.
If we expand this out, we begin to notice a pattern:

$\sum_{k=0}^{n} P[R = k]k^2 =$
$P[R = 1] + P[R = 2] + P[R = 3] + P[R = 4]...$

$$
\begin{aligned}
&+P[R=2]+P[R=3]+P[R=4]...\\
&+P[R=2]+P[R=3]+P[R=4]...\\
&+P[R=2]+P[R=3]+P[R=4]...\\
&\qquad\qquad+P[R=3]+P[R=4]...\\
&\qquad\qquad+P[R=3]+P[R=4]...\\
&\qquad\qquad+P[R=3]+P[R=4]...\\
&\qquad\qquad+P[R=3]+P[R=4]...\\
&\qquad\qquad+P[R=3]+P[R=4]...\\
&\qquad\qquad\qquad\qquad+P[R=4]...\\
&\qquad\qquad\qquad\qquad+P[R=4]...\\
&\qquad\qquad\qquad\qquad+P[R=4]...\\
&\qquad\qquad\qquad\qquad+P[R=4]...\\
&\qquad\qquad\qquad\qquad+P[R=4]...
\end{aligned}
$$

$$= (1^2-0^2)P[R\geq 1]+(2^2-1^2)P[R\geq 2]+(3^2-2^2)P[R\geq 3]+(4^2-3^2)P[R\geq 4]...$$

This pattern and its continuation should be clear. Using this, we can now write $E[R^2]$ as a function of $P[R>k]$, which we computed in part 1:

$$
\begin{aligned}
&E[R^2]=\\
&=\sum_{k=1}^{n} P[R\geq k](k^2-(k-1)^2)=\sum_{k=1}^{n} P[R\geq k](k^2-k^2+2k-1)\\
&=\sum_{k=1}^{n} P[R\geq k](2k-1)=\sum_{k=1}^{n} P[R>k-1](2k-1)\\
&=\sum_{k=0}^{n-1} P[R>k](2k+1)\leq\sum_{k=0}^{n} P[R>k](2k+1)\\
&\leq\sum_{k=0}^{n} P[R>k](2k+1)\\
&\leq 2\sum_{k=0}^{n} k\cdot P[R>k]+\sum_{k=0}^{n} P[R>k].
\end{aligned}
$$

We will now compute both terms in the final expression above. Note that because k goes from 0 to n, we can exchange all appearances of k with $n-k$ without changing anything else:

$$
\begin{aligned}
&2\sum_{k=0}^{n} k\cdot P[R>k]\\
&=2\sum_{k=0}^{n}\frac{n!k}{(n-k)!n^k}=2n!\sum_{k=0}^{n}\frac{k}{(n-k)!n^k}=2n!\sum_{k=0}^{n}\frac{(n-k)}{k!n^{n-k}}=2\frac{n!}{n^n}\sum_{k=0}^{n}\frac{(n-k)n^k}{k!}\\
&=2\frac{n!}{n^n}\sum_{k=0}^{n}\frac{n\cdot n^k}{k!}-2\frac{n!}{n^n}\sum_{k=0}^{n}\frac{k\cdot n^k}{k!}=2\frac{n!n}{n^n}\sum_{k=0}^{n}\frac{n^k}{k!}-2\frac{n!}{n^n}\sum_{k=1}^{n}\frac{k\cdot n^k}{k!}+0=2\frac{n!n}{n^n}\sum_{k=0}^{n}\frac{n^k}{k!}-2\frac{n!}{n^n}\sum_{k=1}^{n}\frac{n^k}{(k-1)!}\\
&=2\frac{n!n}{n^n}\sum_{k=0}^{n}\frac{n^k}{k!}-2\frac{n!}{n^n}\sum_{k=0}^{n-1}\frac{n^{k+1}}{k!}=2\frac{n!n}{n^n}\sum_{k=0}^{n}\frac{n^k}{k!}-2\frac{n!n}{n^n}\sum_{k=0}^{n-1}\frac{n^k}{k!}=2\frac{n!n}{n^n}\cdot(\sum_{k=0}^{n}\frac{n^k}{k!}-\sum_{k=0}^{n-1}\frac{n^k}{k!})\\
&=2\frac{n!n}{n^n}\cdot\frac{n^n}{n!}=2n.
\end{aligned}
$$

Now for the second term:

$$
\begin{aligned}
&\sum_{k=0}^{n} P[R>k]\\
&=\sum_{k=0}^{n}\frac{n!}{(n-k)!n^k}=n!\sum_{k=0}^{n}\frac{1}{k!n^{n-k}}=\frac{n!}{n^n}\sum_{k=0}^{n}\frac{n^k}{k!}\leq\frac{n!e^n}{n^n}=\frac{cn^ne^n\sqrt{2\pi n}}{n^ne^n}=c\sqrt{n}\text{ where } c \text{ is some constant.}
\end{aligned}
$$

Note the use of the Taylor series expansion for e^n. Now combining the second term with the first term, we can give an upper bound for $E[R^2]$:

$$
\begin{aligned}
&E[R^2]=2\sum_{k=0}^{n} k\cdot P[R>k]+\sum_{k=0}^{n} P[R>k]\leq 2n+c\sqrt{n}\\
&\to E[R^2]=O(n).
\end{aligned}
$$

Now for the lower bound. From the notes (page 273 of LL), we have a closed form upper bound on the probability that no two people have the same birthday in a set of k people. But this is the same as the event that, after k balls have been drawn out of 365 total distinct balls, no two are the same. Thus, because the problems are fundamentally the same, we can use this closed form lower bound on $P[R > k]$. Take it for granted that:

$P[R > k] \geq ce^{-k(k-1)/2n}$.

Also note that:

$P[R^2 > k^2] = P[R > k]$ for positive k since R is always positive.

Using this with Markov's inequality, we have:

$P[R^2 > k^2] \leq E[R^2]/k^2 \rightarrow P[R > k] \leq E[R^2]/k^2$.

Essentially what this says is if we know $P[R > k]$, then we have implicitly established a lower bound on $E[R^2]/k^2$. So, since we don't really care about constant factors, let us pick a value for $P[R > k]$. Say $P[R > k] = c_1$ such that $0 < c_1 < 1$. Now, using the lower bound established in the notes, we can put a lower bound on k and plug into the equation above. Doing the math:

$P[R > k] = c_1 \rightarrow c_2 e^{-k(k-1)/2n} \leq c_1$ AND $c_1 \leq E[R^2]/k^2$.

And we can use the first equation to bound k and plug into the second to bound $E[R^2]$.

$c_2 e^{-k(k-1)/2n} \leq c_1 \rightarrow e^{-k(k-1)/2n} \leq c_1/c_2 \rightarrow -k(k-1)/2n \leq \ln(c_1/c_2)$
$\rightarrow k(k-1)/2n \geq -\ln(c_1/c_2) \rightarrow k(k-1) \geq 2n\ln(c_2/c_1) \rightarrow k^2 - k \geq 2n\ln(c_2/c_1) \rightarrow k^2 \geq 2n\ln(c2/c1)$
$\rightarrow k \geq \sqrt{n}\sqrt{2\ln(c_2/c_1)}$.

And finally, using $P[R > k] = c_1$ and $k \geq \sqrt{n}\sqrt{2\ln(c_2/c_1)}$ in our original equation from Markov's inequality, we get:

$P[R > k] = c_1 \leq E[R^2]/k^2 \leq E[R^2]/(2n\ln(c_2/c_1))$
$\rightarrow E[R^2] \geq n \cdot 2c_1 \ln(c_2/c_1)$ for some constants $0 < c_1 < 1$ and $c_2 > 0$
$\rightarrow E[R^2] = \Omega(n)$.

Thus, having shown that $E[R^2] = \Omega(n)$ and $E[R^2] = O(n)$, we can conclude that $E[R^2] = \Theta(n)$ and that, therefore, from the definition of Θ:

$c_1 n \leq E[R^2] \leq c_2 n$ for some constants c_1 and c_2 and n sufficiently large,

therefore bounding $E[R^2]$ within constant factors.

ID: 03 **Collaborator #1:** Last Name, First Name
Name: Al-Naji, Nader **Collaborator #2:** Last Name, First Name

Solution to HW 5, Problem 3

COS 340 - Spring 2012

1. Consider a set of positive integers $x_1, ..., x_l$ in $\{1, ..., n\}$ and define the random variable $X = b_1x_1 + ... + b_lx_l$, where the b_i's are independent random variables equal to 0 and 1 with probability $1/2$. Prove that, for any $\lambda > 1$,

$P[|X - E[X]| \geq \lambda n\sqrt{l}/2] \leq 1/\lambda^2$**.**

Let $\sigma = \sqrt{Var[X]}$ and let $c = \lambda n\sqrt{l}/2/\sigma$. If we choose this particular setting for c, then we have $c\sigma = \lambda n\sqrt{l}/2$ and, further, by substituting $c\sigma$ in for $\lambda n\sqrt{l}/2$ and using the Chebyshev inequality, since $\lambda > 1$, it follows that:

$P[|X - E[X]| \geq \lambda n\sqrt{l}/2] = P[|X - E[X]| \geq c\sigma] \leq 1/c^2 = \frac{\sigma^2}{(1/4)\lambda^2 n^2 l} = (1/\lambda^2) \cdot \frac{\sigma^2}{(1/4)n^2 l}$
$\rightarrow P[|X - E[X]| \geq \lambda n\sqrt{l}/2] \leq (1/\lambda^2) \cdot \frac{\sigma^2}{(1/4)n^2 l} \leq (1/\lambda^2)$ IF $\frac{\sigma^2}{(1/4)n^2 l} \leq 1$ ALWAYS.

So now all we have to do is show that $\frac{\sigma^2}{(1/4)n^2 l}$ is always less than or equal to 1 and we will have our proof, since if this coefficient on $(1/\lambda^2)$ is less than or equal to 1, then the whole quantity $((1/\lambda^2)\frac{\sigma^2}{(1/4)n^2 l})$ will be less than or equal to $(1/\lambda^2)$ since $\lambda > 1$.

Now we show that $\sigma^2 = Var[X] \leq (1/4)n^2 l$. Note that since the b_i are independent, the variance of their sum is equal to the sum of their variances.

$Var[X] = Var[\sum_{i=1}^{l} b_i x_i] = \sum_{i=1}^{l} Var[b_i x_i]$ because the b_i are independent of each other
$\rightarrow \sum_{i=1}^{l} Var[b_i x_i] = \sum_{i=1}^{l} x_i^2 Var[b_i] = \sum_{i=1}^{l} x_i^2 Var[b_1]$ since the b_i are identically distributed $\rightarrow Var[b_i] = Var[b_1]\ \forall i$.

Now we compute $Var[b_1]$ to complete the calculation:

$Var[b_1] = E[b_1^2] - E[b_1]^2 = ((1/2)(1^2)) - ((1/2)(1))^2 = 1/2 - 1/4 = 1/4$. And we have:

$Var[X] = (1/4)\sum_{i=1}^{l} x_i^2$.

Finally, because we are told in the beginning that the maximum value each of the x_i can take on is n, we can compute the upper bound of the variance by assuming that all of the x_i take on their maximum value and get:

$Var[X] = (1/4)\sum_{i=1}^{l} x_i^2 \leq (1/4)\sum_{i=1}^{l} n^2 = (1/4)n^2 l$
$\rightarrow \sigma^2 \leq (1/4)n^2 l$.

This is exactly the result we were looking for and it lets us complete the proof. To resummarize:

$P[|X - E[X]| \geq \lambda n\sqrt{l}/2] = P[|X - E[X]| \geq c\sigma] \leq 1/c^2 = \frac{\sigma^2}{(1/4)\lambda^2 n^2 l} = (1/\lambda^2) \cdot \frac{\sigma^2}{(1/4)n^2 l}$
$\rightarrow P[|X - E[X]| \geq \lambda n\sqrt{l}/2] \leq (1/\lambda^2) \cdot \frac{\sigma^2}{(1/4)n^2 l} \leq (1/\lambda^2)$ since $\frac{\sigma^2}{(1/4)n^2 l} \leq 1$ and $\lambda > 1$

$\rightarrow P[|X - E[X]| \geq \lambda n\sqrt{l}/2] \leq (1/\lambda^2)$.

2. A set of positive integers $x_1, ..., x_l$ is said to have distinct sums if all the numbers $\sum_{i \in S} x_i$ are distinct, for all nonempty subsets $S \subset \{1, ..., l\}$. Let $f(n)$ be the maximal value of l such that there

exists a set of l positive integers in $\{1, ..., n\}$ with distinct sums. Prove that

$f(n) < 2\lg n$.

First, note that because l is a set, it cannot contain more than one instance of each number from 1 to n. Thus, the size of l cannot be exceed n and, further, at its maximum size, l will contain one of each of $\{1, ..., n\}$. So the maximum sum that any subset of $\{1, ..., n\}$ can achieve is the sum of all numbers between 1 and n and thus the sum of any subset of $\{1, ..., n\}$ must take on a value between $[1, \sum_{i=1}^{n} i] = [1, (1/2)n(n+1)]$.
But the number of subsets of a set of size l is equal to 2^l. So if l is too large, namely if $2^l > (1/2)n(n+1)$, then, by the pigeonhole principle, there will be some pair of subsets of l that have the same sum. Thus, for a set of size l constructed from the numbers $\{1, ..., n\}$ to have distinct sums, it must satisfy the following. Note that throughout the proof, the fact that all of our quantities are positive is used implicitly in inequalities and such:

$2^l \leq (1/2)n(n+1) \rightarrow l \leq \lg 1/2 + \lg n + \lg(n+1) \rightarrow l \leq \lg n + (\lg(n+1) - 1)$.

Now we just need to show that $(\lg(n+1) - 1) < \lg n$ and we have our proof:

$\lg(n+1) - 1 < \lg n \rightarrow \lg(n+1) - \lg n < 1 \rightarrow \lg((n+1)/n) < 1 \rightarrow (n+1)/n < 2 \rightarrow n+1 < 2n$
$\rightarrow n < 2n - 1 \rightarrow n < n(2 - 1/n) \leftarrow$ true if $(2 - 1/n) > 1$ since n is positive and coefficient on n being greater than 1 will make the quantity $n(2 - 1/n)$ greater than n. So showing it:
$\rightarrow (2 - 1/n) > 1 \rightarrow 2 > 1 + 1/n \rightarrow 1 > 1/n$
$\rightarrow n > 1$.

So as long as n is greater than 1, a safe assumption, the inequality $\lg(n+1) - 1 < \lg n$ holds and we can finish our proof:

$l \leq \lg n + (\lg(n+1) - 1) < \lg n + \lg n$
$\rightarrow l < 2\lg n$.

ID: 03 **Collaborator #1:** Last Name, First Name
Name: Al-Naji, Nader **Collaborator #2:** Last Name, First Name

Solution to HW 5, Problem 4

COS 340 - Spring 2012

TilghmanPCS has introduced a new pay-as-you-go data plan for Princeton: Customers can buy different kinds of bundles every month (and several in a month), where each bundle has a fixed price and buys them the ability to transfer a certain number of megabytes in the month that the bundle is bought. Unused transfer capacity expires at the end of the month. Tilghman offers a choice of bundles labelled type 1, type 2, ..., type i for every integer $i > 0$. A type i bundle costs i dollars upfront and allows the customer to transfer i^2 megabytes in a month. Bundles can be bought at any time in the month (and the same bundle can be bought repeatedly if required) - additional bundles add to the existing capacity already purchased. However, unused megabytes expire at the end of the month. Your goal is to design an online algorithm for the customer to determine which bundles to purchase based on the current usage. Assume that whenever the current transfer capacity is exhausted, the customer has to decide which bundle to purchase for additional data transfers.

1. Determine the optimal offline solution for this problem.

The optimal offline algorithm knows, a priori, exactly how many megabytes a user will require during the month. Note that we hereafter refer to packages as the individual type i items a user can purchase and refer to bundles as being made up of one or more packages. Let the number of megabytes the user requires in a particular month be denoted by n and let the amount of money the offline algorithm decides to spend be denoted by k. Let V_k be the set of all possible numbers of megabytes that can be purchased with k dollars. So, $V_1 = \{1\}$, for example, since 1 dollar can only purchase exactly 1 megabyte. $V_2 = \{2, 4\}$, on the other hand, since a bundle costing 2 dollars that consists of 2 type 1 packages gets the user 2 megabytes, and a bundle costing 2 dollars consisting of 1 type 2 package gets the user 4 megabytes. With this notation down, it is the job of the optimal offline algorithm to find a bundle b of cost k such that:

$\sup(V_{k-1}) < n$ and (number of megabytes purchased by bundle b)$\geq n$.

In words, the number of megabytes purchased by bundle b must be at least n and, if bundle b costs k dollars, there must not exist a bundle costing $k-1$ dollars that can also purchase n megabytes.

The most cost-effective way to purchase at least n megabytes is to buy a type $\lceil\sqrt{n}\rceil$ bundle. We will now show that this is the case by proving that no bundle costing k dollars of consisting of more than one package can purchase more megabytes than a single type k package.

Consider a bundle costing k dollars consisting of a single package and another bundle consisting of two packages costing i dollars and j dollars respectively such that $i + j = k$. We want to show that the number of megabytes purchased by the first bundle is more than the number of megabytes purchased by the second bundle for all choices of $i, j | i + j = k$. So, we show:

$k^2 > i^2 + j^2 \ \forall i, j \mid i + j = k$
$\rightarrow k^2 > i^2 + (k - i)^2$
$\rightarrow k^2 > i^2 + k^2 - 2ki + i^2$
$\rightarrow k^2 > k^2 - 2(ki - i^2)$
Note that $ki > i^2$ since we defined i to be strictly less than k and implies $\rightarrow (ki - i^2) > 0$
$\rightarrow k^2 > k^2 - 2$(some positive value); a true statement, which implies

$\rightarrow k^2 > i^2 + j^2 \ \forall i, j \mid i + j = k$ is true.

Having shown this inequality, we now interpret its result. This inequality shows that any bundle costing k dollars that consists of more than one package will purchase strictly less than a bundle consisting of a single package that costs k dollars. This result shows that breaking a single package up into multiple packages lowers the buying power of that package in general. So a bundle costing k dollars consisting of a single package will purchase more megabytes than a bundle consisting of $i > 1$ packages in general since breaking up the single package causes its buying power and, consequently, the buying power of the whole bundle, to decrease.

This result serves to show that the most cost-effective bundle that can purchase n megabytes when n is a perfect square is a type $\sqrt{n}$ bundle since every other bundle consisting of more than a single package will purchase strictly fewer megabytes than this bundle, and therefore not satisfy the user. When n is not a perfect square, we know that $n > (\lfloor\sqrt{n}\rfloor)^2$ so a bundle costing $\lfloor\sqrt{n}\rfloor$ cannot purchase enough megabytes to satisfy the user's need. Further, using our result from above, because this single-package bundle costing $\lfloor\sqrt{n}\rfloor$ dollars cannot purchase n megabytes, we know that NO bundle costing $\lfloor\sqrt{n}\rfloor$ dollars can purchase n megabytes and so the best we can do is to buy a bundle costing $\lceil\sqrt{n}\rceil = \lfloor\sqrt{n}\rfloor + 1$ dollars since n is not a perfect square. While it isn't always strictly necessary that this bundle be a single-package type $\lceil\sqrt{n}\rceil$ bundle, we may as well purchase this particular bundle in the case that n is not a perfect square because we don't care about wasted bandwidth. For example, if $n = 82$, we could buy a type 10 bundle, or a type 9 and type 1 bundle, but we choose the type 10 bundle in this case.

To summarize the optimal offline algorithm:
Always purchase a single type $\lceil\sqrt{n}\rceil$ package at the beginning of the month for a cost of $\lceil\sqrt{n}\rceil$ dollars.

2. Design and analyze an online algorithm with constant competitive ration.

We will analyze the following algorithm:

1. Start out each month with $i = 0$ and purchase a single type 2^i package.
2. If/when this this package gets used up, increment i (so $i = i + 1$) and purchase a type 2^i package.

So the first package purchased will be of type 1. If the user exhausts this package, they will purchase a type 2 package, then if that gets used up a type 4 package, and so on. In order to compute this package's cost as a function of n, we note two facts. In the following, l is the number of times a user exhausts a package:
$k = \sum_{i=0}^{l} 2^i$ and $n \geq \sum_{i=0}^{l-1} 2^{2i}$.
The first fact is true because if we run out l times, it means we bought $l + 1$ packages each costing 2^i for i from 0 to l.
The second fact is true because in order for us to have purchased our last package, we will have to have run out when using our second to last package and so n must be greater than the sum of our package values if we don't include the value of the last package we bought.

We will now solve for l in the second inequality and plug it into the first equality in order to get k as a function of l.

$n \geq \sum_{i=0}^{l-1} 2^{2i}$
$\rightarrow n \geq (1 - 4^l)/(1 - 4)$
$\rightarrow n \geq (4^l - 1)/(3)$
$\rightarrow 3n \geq 4^l - 1 \rightarrow 3n \geq 4^l$
$\rightarrow \lg(3n) \geq l \cdot \lg 4 \rightarrow \lg(3n) \geq l \cdot 2$

$\rightarrow l \leq \frac{\lg(3n)}{2}$

Plugging this into the first equation, we have:

$k = \sum_{i=0}^{l} 2^i = (1 - 2^{l+1})/(1 - 2) = (2^{l+1} - 1)/(1)$
$\rightarrow k = 2 \cdot 2^l - 1 \leq 2 \cdot 2^{\frac{\lg(3n)}{2}} - 1 = 2 \cdot (2^{\lg(3n)})^{1/2} - 1 = 2 \cdot \sqrt{3n} - 1$

$\rightarrow k \leq 2\sqrt{3n}$

Finally having bounded k for our online algorithm, we can compare the performance of the online algorithm to the offline algorithm. Note that it doesn't matter how quickly or how slowly the user uses up megabytes, only how many, denoted by n. So we have:

$C_{opt}[n] = \lceil \sqrt{n} \rceil \geq \sqrt{n}$ and
$C_A[n] \leq 2\sqrt{3n}$. So,
$C_A/C_{opt} \leq \frac{2\sqrt{3n}}{\sqrt{n}} = 2\sqrt{3}$.

So this algorithm is $(2\sqrt{3})$-competitive.