TODO: OK so actually Lemma 1 needs to take place in the S-skip-emptyings, E-extra-emptyings negative-fill variable-processor cup game. And I can pretty much just be like yo, if the emptier skips or does extra emptyings globally on AB, then I just give up on everything besides keeping the cups $(R_\Delta+2h)$-flat. I've modified the lemma statement to this effect, but not the proof yet.

Now we show that we can force a constant fraction of the cups to have high fill; using Lemma 1 and exploiting the greedy-like nature of the emptier we can get a known cup with high fill (we show this in Proposition 1).

Lemma 1. *Let $\Delta \le O(1)$, let $h \le O(1)$ with $h \ge 16+16\Delta$, let n be at least a sufficiently large constant determined by h and Δ, and let $R \le poly(n)$. Consider an R-flat cup configuration in the S-skip-emptyings E-extra-emptyings variable-processor cup game on n cups. Let A, B be disjoint subsets of the cups with $|AB| = n$. Over the course of the algorithm B will give some cups to A, but $|A|$ will always satisfy $|A| \ll |B|$, and $|A|$ will eventually be $\Theta(n)$. Let $M \gg n$ be very large.*

There is an oblivious filling strategy that, conditional on the emptier not using extra-emptying, either achieves mass at least M in the cups, or makes an unknown set of $\Theta(n)$ cups in A have fill at least h with probability at least $1-2^{-\Omega(n)}$ in running time poly(M) against a Δ-greedy-like emptier while also guaranteeing that $\mu(B) \ge -h/2$.

Furthermore, this filling strategy guarantees that the cups are $(R_\Delta+2h)$-flat on every round.

Proof. For now we condition on the emptier not using extra emptying and show that we can achieve the desired result in this case (good backlog with exponentially good probability); at the end of the proof we will consider the case where the emptier does use extra emptyings, and bound the fill-range in that case.

We refer to A as the ***anchor*** set, and B as the ***non-anchor*** set. Let $n_A = \Theta(n)$ be small enough to satisfy

$$n_A \le (n-n_A)/(2e^{2h+1}+1). \tag{1}$$

The filler initializes A to $\varnothing$, and B to be all of the cups. Over the course of the algorithm B will give away n_A cups to A. Note that $|B| \ge n-n_A \gg n_A \ge |A|$.

We denote by *randalg* the oblivious filling strategy given by **??**. We denote by *flatalg* the oblivious filling strategy given by **??**. We say that the filler ***applies*** a filling strategy *alg* to a set of cups $D \subseteq B$ if the filler uses *alg* on D while placing 1 unit of fill in each anchor cup.

We now describe the filler's strategy.

The filler starts by flattening the cups, i.e. using *flatalg* on all of the cups for poly(M) rounds (setting $p = n/2$). After this the filling strategy always places 1 unit of water into each anchor cup on every round. The filler performs a series of n_A ***donation-processes***, which are procedures that the filler uses to get a new cup—which will sometimes have high fill—in the anchor set. On each donation-process the filler applies *randalg* many times to arbitrarily chosen constant-size sets $D \subset B$ with $|D| = \lceil e^{2h+1} \rceil$. The number of times that the filler applies *randalg* is chosen at the start of the donation-process, chosen uniformly at random from $[m]$ ($m = \text{poly}(M)$ to be specified). At the end of each donation-process, the filler does a ***donation***: the filler takes the cup given by *randalg* in B evicts it from B and adds it to A. After performing a donation the filler must increase p by 1 so that $p = |A|+1$. Before each application of *randalg* the filler flattens B by applying *flatalg* to B for poly(M) rounds.

We remark that this construction is similar to the construction in **??**, but has a major difference that substantially complicates the analysis: in the adaptive lower bound construction the filler halts after achieving the desired average fill in the anchor set, whereas the oblivious filler cannot halt but rather must rely on the emptier's greediness to guarantee that each application of *randalg* has constant probability of generating a cup with high fill.

We proceed to analyze our algorithm.

Note that if the emptier skips more than M emptyings, or neglects the anchor set more than M times without decreasing the fills of any anchor cups in between these times, then the filler has achieved mass M in the cups, in which case the filler has fulfilled the statement of Lemma 1. Throughout the remainder of the analysis we consider the case of an emptier that chooses to not skip more than M times, and chooses to not neglect the anchor set more than M times without decreasing the fill of anchor cups in between these times.

First note that the initial flattening makes the cups R_Δ-flat by **??**. In particular, note that the flattening happens in the $(n/2)$-processor M-extra-emptyings M-skip-emptyings variable-processor cup game on n cups.

We say that a property of the cups has ***always*** held if the property has held since the start of the first donation-process; i.e. from now on we only consider rounds after the initial flattening.

We say that the emptier ***neglects*** the anchor set on a round if it does not empty from each anchor cup. We say that an application of *randalg* to $D \subset B$

is ***non-emptier-wasted*** if the emptier does not neglect the anchor set during any round of the application of *randalg*. We define $d = \sum_{i=2}^{|D|} 1/i$ (recall that $|D| = \lceil e^{2h+1} \rceil$). We say that an application of *randalg* to D is ***lucky*** if it achieves backlog at least $\mu(B) - R_\Delta + d$; note that by **??** if we condition on an application of *randalg* where B started R_Δ-flat being non-emptier-wasted then the application has at least a $1/|D|!$ chance of being lucky.

Now we prove several important bounds on fills of cups in A and B.

Claim 1. *All applications of flatalg make B be R_Δ-flat and B is always $(R_\Delta + d)$-flat.*

Proof. Given that the application of *flatalg* immediately prior to an application of *randalg* made B be R_Δ-flat, by **??** we have that B will stay $(R_\Delta + d)$-flat during the application of *randalg*. Given that the application of *randalg* immediately prior to an application of *flatalg* resulted in B being $(R_\Delta + d)$-flat, we have that B remains $(R_\Delta + d)$-flat throughout the duration of the application of *flatalg* by **??**. Given that B is $(R_\Delta + d)$-flat before a donation occurs B is clearly still $(R_\Delta + d)$-flat after the donation, because the only change to B during a donation is that a cup is removed from B which cannot increase the fill-range of B. Note that B started R_Δ-flat before the first application of *flatalg* because all the cups was flattened. Note that if an application of *flatalg* begins with B being $(R_\Delta + d)$-flat, then by considering the flattening to happen in the $(|B|/2)$-processor M-extra-emptyings M-skip-emptyings cup game we ensure that it makes B be R_Δ-flat, because the emptier cannot skip more than M emptyings and also cannot do more than M extra emptyings or the mass would be at least M. Hence we have by induction that B has always been $(R_\Delta + d)$-flat and that all flattening processes have made B be R_Δ-flat. ☐

Now we aim to show that $\mu(B)$ is never too low, which we need in order to establish that every non-emptier-wasted lucky application of *randalg* gets a cup with high fill. Interestingly in order to lower bound $\mu(B)$ we first must upper bound $\mu(B)$, which by greediness and flatness of B gives an upper bound on $\mu(A)$ which we use to get a lower bound on $\mu(B)$.

Claim 2. *We have always had*

$$\mu(B) \leq 2 + \mu(AB).$$

Proof. There are two ways that $\mu(B) - \mu(AB)$ can increase:
Case 1: The emptier could empty from 0 cups in B while emptying from every cup in A.
Case 2: The filler could evict a cup with fill lower than $\mu(B)$ from B at the end of a donation-process.

Note that cases are exhaustive, in particular note that if the emptier skips more than 1 emptying then $\mu(B) - \mu(AB)$ must decrease because $|A| \approx |AB|$, in particular (1), as opposed to in Case 1 where $\mu(B) - \mu(AB)$ increases.

In Case 1, because the emptier is Δ-greedy-like,

$$\min_{a \in A} \text{fill}(a) > \max_{b \in B} \text{fill}(b) - \Delta.$$

Thus $\mu(B) \leq \mu(A) + \Delta$. As $|B| \gg |A|$, in particular by (1), this can be loosened to $\mu(B) \leq 1 + \mu(AB)$.

Consider the final round on which B is skipped while A is not skipped (or consider the first round if there is no such round).

From this round onwards the only increase to $\mu(B) - \mu(AB)$ is due to B evicting cups with fill well below $\mu(B)$. We can upper bound the increase of $\mu(B) - \mu(AB)$ by the increase of $\mu(B)$ as $\mu(AB)$ is strictly increasing.

The cup that B evicts at the end of a donation-process has fill at least $\mu(B) - R_\Delta - (|D| - 1)$, as the running time of *randalg* is $|D| - 1$, and because B starts R_Δ-flat by Claim 1. Evicting a cup with fill $\mu(B) - R_\Delta - (|D| - 1)$ from B changes $\mu(B)$ by $(R_\Delta + |D| - 1)/(|B| - 1)$ where $|B|$ is the size of B before the cup is evicted from B. Even if this happens on each of the n_A donation-processes $\mu(B)$ cannot rise higher than $n_A(R_\Delta + |D| - 1)/(n - n_A)$ which by design in choosing $|B| \gg |A|$, as was done in (1), is at most 1.

Thus it always is the case that $\mu(B) \leq 2 + \mu(AB)$. ☐

The upper bound on $\mu(B)$ along with the guarantee that B is flat allows us to bound the highest that a cup in A could rise by greediness, which in turn upper bounds $\mu(A)$ which in turn lower bounds $\mu(B)$. In particular we have

Claim 3. *We always have*

$$\mu(B) \geq -h/2.$$

Proof. By Claim 2 and Claim 1 we have that no cup in B ever has fill greater than $u_B = \mu(AB) + 2 + R_\Delta + d$.

Let $u_A = u_B + \Delta + 1$. We claim that the backlog in A never exceeds u_A.

Consider how high the fill of a cup $c \in A$ could be. If c came from B then when it is donated to A its fill is at most $u_B < u_A$. Otherwise, c started with fill at most $R_\Delta < u_A$. Now consider how much the fill of c

could increase while being in A. Because the emptier is Δ-greedy-like, if a cup $c \in A$ has fill more than Δ higher than the backlog in B then c must be emptied from, so any cup with fill at least $u_B + \Delta = u_A - 1$ must be emptied from, and hence u_A upper bounds the backlog in A.

Of course an upper bound on backlog in A also serves as an upper bound on the average fill of A as well, i.e. $\mu(A) \le u_A$. Rearranging the expression

$$|B|\mu(B) + |A|\mu(A) = |AB|\mu(AB)$$

we have

$$\begin{aligned}
\mu(B)& \\
&= -\frac{|A|}{|B|}\mu(A) + \frac{|AB|}{|B|}\mu(AB) \\
&\ge -(\mu(AB) + 3 + R_\Delta + d + \Delta)\frac{|A|}{|B|} + \frac{|AB|}{|B|}\mu(AB) \\
&= -(3 + R_\Delta + d + \Delta)\frac{|A|}{|B|} + \mu(AB) \\
&\ge -h/2
\end{aligned}$$

where the final inequality follows because $\mu(AB) \ge 0$, and $|B| \gg |A|$, in particular by (1).

$\square$

Now we show that we can at least a constant fraction of the donation-processes succeed with exponentially good probability.

Claim 4. *There exists choice of $m = poly(M)$ such that with probability at least $1 - 2^{-\Omega(n)}$, the filler achieves fill at least h in at least $\Theta(n)$ of the cups in A.*

Proof. If the emptier was not allowed to neglect the anchor set ever and use extra-emptyings in B then the claim would be true as each application of *randalg* would unconditionally succeed with constant probability, so a Chernoff bound would give that $\Theta(n)$ of the donation-processes donate a cup with fill at least $\mu(B) - R_\Delta + d \ge h$, where the inequality follows from Claim 3 which asserts that $\mu(B) \ge -h/2$, and from the facts $d \ge 2h$ and $h \ge 16(1+\Delta)$. However, the emptier is allowed to neglect the anchor set, and in fact the emptier can choose to neglect the anchor set conditional on the filler's progress during *randalg*.

We can lower bound the probability of getting $\Theta(n)$ cups with fills all at least h by considering an augmented emptier that is allowed to interfere with M applications of *randalg* per donation-process that only interferes with applications of *randalg* that would otherwise donate a cup with fill h into A. The optimal strategy for such an emptier, given our filler's strategy of randomly choosing which application to donate a cup on, is to simply interfere with the first M applications of *randalg* that without interference would have achieved a cup with fill h. The filler sets $m = 4M|D|!$. Conditional on the emptier not interfering, each of these applications of *randalg* has at least a $1/|D|!$ chance of getting a cup with fill h. Hence, by a Chernoff bound with exponentially good probability at least $2M$ of the applications of *randalg* have the potential to donate a cup with fill h to A, if the emptier does not interfere. The filler chooses an application uniformly at random from all m applications on which to donate a cup. With probability at least $1/|D|!$ this is on an application where the filler could get a cup with fill h in A if the emptier does not interfere, and with probability at least $1/2$ the emptier does not interfere on this application of *randalg*, because the emptier can interfere on at most M of the applications of *randalg*.

Against this augmented emptier whether or not donation-processes achieve a cup with fill h in A are independent events. As each happens with at least constant probability, by a Chernoff bound there is exponentially high probability that at least a constant fraction of them succeed.

Note that we used the Chernoff bound $\Theta(n)$; by a union bound there is exponentially good probability that all of the desired events occur.

$\square$

We now analyze the running time of the filling strategy. There are $|A|$ donation-processes. Each donation-process consists of poly(M) applications of *randalg*, which each take constant time, and poly(M) applications of *flatalg*, which each take poly(M) time. Thus overall the algorithm takes poly(M) time, as desired.

TODO: Finally we consider the case where the emptier does use extra-emptyings; here all we aim to show is a bound on the fill-range.

$\square$

Finally, using Lemma 1 we can show in Proposition 1 that an oblivious filler can achieve constant backlog. We remark that Proposition 1 plays a similar role in the proof of the lower bound on backlog as **??** does in the adaptive case, but is vastly more complicated to prove (in particular, **??** is trivial, whereas we have already proved several lemmas and propositions as preparation for the proof of Proposition 1).

Proposition 1. *Let $H \le O(1)$, let $\Delta \le O(1)$, let n be at least a sufficiently large constant determined by H and Δ, and let $R \le poly(n)$. Let $M \gg n$ be very large. Consider an R-flat cup configuration in the* *negative-fill variable-processor cup game on n cups with average fill 0. Given this configuration, an oblivious filler can either achieve mass M among the cups, or achieve fill H in a chosen cup in running time poly(M) against a Δ-greedy-like emptier with probability at least $1 - 2^{-\Omega(n)}$.*

Proof. The filler starts by performing the procedure detailed in Lemma 1, using $h = H \cdot 16(1 + \Delta)$. Let the number of cups which must now exist with fill h be of size $nc = \Theta(n)$.

The filler reduces the number of processors to $p = nc$. Now the filler exploits the filler's greedy-like nature to to get fill H in a set $S \subset B$ of nc chosen cups.

The filler places 1 unit of fill into each cup in S. Because the emptier is greedy-like it must focus on the nc cups in A with fill at least h until the cups in S have sufficiently high fill. In particular, $(5/8)h$ rounds suffice. Over $(5/8)h$ rounds the nc high cups in A cannot have their fill decrease below $(3/8)h \geq h/8 + \Delta$. Hence, any cups with fills less than $h/8$ must not be emptied from during these rounds. The fills of the cups in S must start as at least $-h/2$ as $\mu(B) \geq -h/2$. After $(5/8)h$ rounds the fills of the cups in S are at least $h/8$, because throughout this process the emptier cannot have emptied from them until they got fill at least $h/8$, and if they are never emptied from then they achieve fill $h/8$.

Thus the filling strategy achieves backlog $h/8 \geq H$ in some known cup (in fact in all cups in S, but a single cup suffices), as desired.

□

Next we prove the ***Oblivious Amplification Lemma***. The same idea of using a function multiple times on subsets of the cups drives both the Lemma 2 and **??**; however the Oblivious Amplification Lemma is more difficult to prove.

Lemma 2 (Oblivious Amplification Lemma)**.** *Let $0 < \delta \ll 1/2, 1/2 \ll \phi < 1$ be constant parameters, and let $\eta \in \mathbb{N}$ be a function of ϕ. Let $\Delta \leq O(1)$, $R, R' \geq R_\Delta$. Let $\operatorname{alg}(f)$ be an oblivious filling strategy that, conditional on the emptier not using extra-emptyings, either achieves mass M or achieves backlog $f(n)$ in the S-skip-emptyings E-extra-emptyings negative-fill variable-processor cup game on n cups with probability at least $1 - 2^{-\Omega(n)}$ in running time $T(n) \leq poly(n)$ when given a R-flat cup configuration against a Δ-greedy-like emptier, where $M \gg n$ is very large. Furthermore, let $\operatorname{alg}(f)$ guarantee that the cups are $(R_\Delta + f(n))$-flat.*

There exists an oblivious filling strategy $\operatorname{alg}(f')$ that, conditional on the emptier not using extra-emptyings, either achieves mass M' or achieves backlog $f'(n)$ satisfying

$$f'(n) \geq (1-\delta)(\phi - 1/(\delta n))(f(\lfloor (1-\delta)n \rfloor) - R_\Delta) + f(\lceil \delta n \rceil)$$

and $f'(n) \geq f(n)$, in the S'-skip-emptyings E'-extra-emptyings negative-fill variable-processor cup game on n cups with probability at least $1 - 2^{-\Omega(n)}$ in running time

$$T'(n) \leq O(M') + 6\delta n^{\eta+1} T(\lfloor (1-\delta)n \rfloor) + T(\lceil \delta n \rceil)$$

when given a M'-flat cup configuration against a Δ-greedy-like emptier, where $M \gg n$ is very large. Furthermore, $\operatorname{alg}(f')$ guarantees that the cups are always $(R_\Delta + f'(n))$-flat.

Proof. For now we condition on the emptier not using extra emptying and show that we can achieve the desired result in this case (good backlog with exponentially good probability); at the end of the proof we will consider the case where the emptier does use extra emptyings, and bound the fill-range in that case.

The algorithm defaults to using $\operatorname{alg}(f)$ on all the cups if

$$f(n) \geq (1-\delta)(\phi - 1/(\delta n))(f(\lfloor (1-\delta)n \rfloor) - R_\Delta) + f(\lceil \delta n \rceil)$$

In this case our strategy trivially results in the desired backlog in the desired running time. In the rest of the proof we consider the case where we cannot simply fall back on $\operatorname{alg}(f)$ to achieve the desired backlog.

We refer to A as the ***anchor*** set and B as the ***non-anchor*** set. Let $n_A = \lceil \delta n \rceil$, $n_B = \lfloor (1-\delta)n \rfloor$. The filler initializes A to $\varnothing$, and B to be all of the cups. Over the course of $\operatorname{alg}(f')$ B will donate n_A cups to B; note that we always have $|B| \geq n_B$, $|A| \leq n_A$ with equality achieved after n_A donations.

We denote by *flatalg* the oblivious filling strategy given in **??**. We say that the filler ***applies*** an algorithm *alg* to B if it uses *alg* on B while placing 1 unit of fill in each anchor cup.

We now describe the filler's strategy.

At a high level the filler's strategy is as follows:

Step 1: Using $\operatorname{alg}(f)$ repeatedly on B, achieve a cup with fill $\mu(B) + f(|B|)$ in B and then donate this cup into A.

Step 2: Use $\operatorname{alg}(f)$ once on A to obtain a cup in A with fill $\mu(A) + f(|A|)$.

We now describe in detail how to achieve Step 1, which is complicated by the fact that the emptier may attempt to prevent the filler from achieving high fill in a cup in B, and further by the fact that the filler, being oblivious, cannot know if the emptier has done this. In particular, some applications of $\operatorname{alg}(f)$ may fail, but we show that with exponentially good probability a $(1-\phi)$-fraction of the applications of $\operatorname{alg}(f)$ succeed.

The filler starts by flattening the cups, i.e. using *flatalg* on all of the cups for $\operatorname{poly}(M')$ rounds (setting $p = n/2$). After this the filling strategy always places 1 unit of fill into each anchor cup on every round. The filler performs a series of n_A ***donation-processes***, which are procedures that the filler uses to get a new cup—which will sometimes have high fill—in the anchor set. On each donation-process the filler applies $\operatorname{alg}(f)$ many times to B. The number of times that the filler applies *randalg* is chosen at the start of the donation-process, chosen uniformly at random from $[m]$ ($m = \operatorname{poly}(M)$ to be specified). At the end of each donation-process, the filler does a ***donation***: the filler takes the cup given by $\operatorname{alg}(f)$ in B, evicts it from B and adds it to A. After performing a donation the filler must increase p by 1 so that $p = |A| + 1$. Before each application of $\operatorname{alg}(f)$ the filler flattens B by applying *flatalg* to B for $\operatorname{poly}(M)$ rounds.

We proceed to analyze our algorithm.

TODO: For each cup in A the filler performs a procedure called a ***swapping-process***. Let A_0 be initialized to $\varnothing$; during each swapping-process the filler will get some cup in B to have high fill (with very good probability), and then swap this cup into A, and place the cup in A_0 too. We say that the filler ***applies*** $\operatorname{alg}(f)$ to B if it follows the filling strategy $\operatorname{alg}(f)$ on B while placing 1 unit of fill in each anchor cup; during a swapping-process the filler repeatedly applies $\operatorname{alg}(f)$ to B, flattening $B \cup (A \setminus A_0)$, which results in B being R_Δ-flat as well, before each application. We say that the emptier ***neglects*** the anchor set on a round if the emptier does not empty from every anchor cup on this round. The mass of the anchor set increases by at least 1 each round that the anchor set is neglected. An application of $\mathrm{alg}(f)$ to B is said to be ***successful*** if A is never neglected during the application of $\mathrm{alg}(f)$ to B. We say that a swapping-process is ***successful*** if the application of $\mathrm{alg}(f)$ on which the filler swaps a cup into A is a successful application of $\mathrm{alg}(f)$.

TODO: Let $\mu_\Delta = 2R_\Delta + \Delta$; the emptier, being Δ-greedy-like, cannot neglect the anchor set more than $n\delta\mu_\Delta$ times. Thus, by making each swapping-process consist of n^η applications of $\mathrm{alg}(f)$ to B and then choosing a single application among these (uniformly at random) after which to swap a cup into A (and we also place the cup in A_0; A_0 consists of all cups in A that were swapped into A from B), we guarantee that with probability at least $n\delta\mu_\Delta/n^\eta$ this swap occurs at the end of a successful application of $\mathrm{alg}(f)$ to B.

If an application of $\mathrm{alg}(f)$ is successful, then with probability at least $1-2^{-\Omega(n)}$ it generates a cup with fill $f(|B|)+\mu(B)$ in B, because equal resources were put into B on each round while $\mathrm{alg}(f)$ was used, and the cup state started as R_Δ-flat and hence also started as M-flat (as $M \geq R_\Delta$).

Now we aim to show that $\mu(A)$ is large; we do so by showing that $\mu(B)$ is small (i.e. very negative). Because the probability of an application of $\mathrm{alg}(f)$ being successful is only $1-1/\mathrm{poly}(n)$, which is in particular not as good as the $1-2^{-\Omega(n)}$ that we will guarantee, we will not be able to actually assume that every such application of $\mathrm{alg}(f)$ is successful. However, (as we will show later) we can guarantee that at least a constant fraction ϕ of the swapping-processes are successful with exponentially good probability.

The filler swaps $|A|$ cups into B. Consider how $\mu(B\cup A\setminus A_0)$ changes when a new cup is swapped into A and placed in A_0. Let the initial value of $\mu(B\cup A\setminus A_0)$ be μ_0. Say that initially $|A_0| = i$ (i.e. i swapping-processes have occured so far). If the swapping-process is successful then the swapped cup has fill at least $\mu_0 - R_\Delta + f(|B|)$. Hence the new average fill of $B\cup A\setminus A_0$ after the swap is

$$\frac{\mu_0\cdot(n-i)-(\mu_0-R_\Delta+f(|B|))}{n-i-1} = \mu_0 - \frac{f(|B|)-R_\Delta}{n-i-1}.$$

This recurrence relation allows us to find the value of $\mu(B\cup A\setminus A_0) = \mu(B)$ after $|A|$ swapping processes (i.e. once $A\setminus A_0 = \varnothing$):

$$\mu(B) \leq -\sum_{i=1}^{|A|\phi} \frac{f(|B|)-R_\Delta}{n-i}.$$

Now we bound $H_{n-1} - H_{n-|A|\phi-1}$ where H_i is the i-th harmonic number. Using the fact that

$$H_n = \ln n + \gamma + 1/(2n) - 1/(12n^2) + 1/(120n^4) - \dots$$

we have,

$$\begin{aligned}
&H_{n-1} - H_{n-|A|\phi-1} \\
&\geq \ln\frac{n-1}{n-|A|\phi-1} - \frac{1}{2(n-|A|\phi-1)} \\
&\geq \ln\frac{n}{n-|A|\phi} - \frac{1}{n} \\
&= \ln\frac{n}{n-\lceil\delta n\rceil\phi} - \frac{1}{n} \\
&\geq \ln\frac{1}{1-\delta\phi} - \frac{1}{n} \\
&\geq \delta\phi - \frac{1}{n}.
\end{aligned}$$

Hence we have,

$$\mu(A) \geq \frac{(1-\delta)}{\delta}\left(\delta\phi - \frac{1}{n}\right)(f(|B|)-R_\Delta). \tag{2}$$

Now we establish that we can guarantee that $\phi|A|$ of the $|A|$ swapping-process succeed for any choice of $\phi = \Theta(1)$ by sufficiently large choice of η, i.e. by performing enough applications of $\mathrm{alg}(f)$ within each swapping-process. Recall that by construction of μ_Δ the emptier cannot neglect the anchor set on more than $n\delta\mu_\Delta$ applications of $\mathrm{alg}(f)$ to B.

Let X_i be the random variable that indicates the event that the i-th swapping-process was not successful; note that the X_i are independent, because the filler's random choices of which applications of $\mathrm{alg}(f)$ within each swapping-process on which to swap a cup into the anchor set are independent. We have, for any constant ϕ,

$$\begin{aligned}
\Pr\left[\left|\frac{1}{|A|}\sum_{i=1}^{|A|} X_i - \frac{n\delta\mu_\Delta}{n^\eta}\right| \geq 1-2\phi\right] &\leq 2e^{-2|A|(1-2\phi)^2} \\
&\leq 2^{-\Omega(n)}.
\end{aligned}$$

By appropriately large choice for $\eta \leq O(1)$,

$$n\delta\mu_\Delta/n^\eta \leq \phi$$

no matter how small $w \geq \Omega(1)$ is chosen. In particular this implies that

$$\Pr\left[\sum_{i=1}^{|A|} X_i \geq |A|(1-\phi)\right] \geq 1 - 2^{\Omega(n)}.$$

That is, with exponentially good probability $|A|\phi$ of the swapping processes succeed. Taking a union bound over all applications of alg(f) we have that there is exponentially good probability that all applications of alg(f) succeeded. Thus, with exponentially good probability, by (2), Step 1 achieves backlog

$$(1-\delta)(\phi - 1/(\delta n)(f(\lfloor(1-\delta)n\rfloor - R_\Delta)$$

To achieve Step 2 the filler simply applies alg(f) to A. This clearly achieves backlog

$$f(|A|) = f(\lceil \delta n \rceil)$$

with exponentially good probability.

Since both Step 1 and Step 2 succeed with exponentially good probability, the entire process succeeds with exponentially good probability.

We now analyze the running time of alg(f'). The initial smoothing takes time $O(M')$. Step 1 entails $n^\eta \cdot (n\delta)$ swapping-processes, each of which takes time $f(|B|)$. Due to flattening at the beginning of each application of alg(f) the running time may be increased by a multiplicative factor of at most 3. Step 2 takes time $T(|A|)$. Adding these times we have that the running time $T'(n)$ of alg(f') is

$$T'(n) \leq O(M') + 6\delta n^{\eta+1} T(\lfloor(1-\delta)n\rfloor) + T(\lceil \delta n \rceil).$$

Having proved that alg(f') achieves the desired backlog with the desired probability in the desired running time, the proof is now complete.

□

References

[1] Michael A Bender, Martín Farach-Colton, and William Kuszmaul. Achieving optimal backlog in multi-processor cup games. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 1148–1157, 2019.

[2] Michael A Bender, Sándor P Fekete, Alexander Kröller, Vincenzo Liberatore, Joseph SB Mitchell, Valentin Polishchuk, and Jukka Suomela. The minimum backlog problem. *Theoretical Computer Science*, 605:51–61, 2015.

[3] Paul Dietz and Rajeev Raman. Persistence, amortization and randomization. 1991.

[4] William Kuszmaul. Achieving optimal backlog in the vanilla multi-processor cup game. *SIAM*, 2020.