

# CMPS 102 — Fall 2018 – Homework 1

Alyssa Melton

I have read and agree to the collaboration policy.

Collaborators: none

## Solution to Problem 3

At a glance, this procedure seems to only make the bag grow in token size, but by paying closer attention to what exactly it is that adds more tokens to the bag makes it very obvious that the growing will soon stop. When pulling two tokens, we have three cases, which we will analyze.

Case 1: If one of the tokens is red, do nothing.

Here, we see that this eliminates two tokens at a time, one being red, the other being any color.

Case 2: If both tokens are green, we put 1 green token and 2 blue tokens back into the bag.

This case only happens if two green tokens are pulled from the bag. Now, the result of this action is to put one of the green tokens back, in addition to two blue tokens. Looking at all the cases, we see that none add green tokens when executed, and since we are always pulling out more green tokens than we are putting back, we will eventually run out of green tokens.

Case 3: If we got one blue token, and the other token is not red, then we put 3 red tokens back into the bag.

This case removes one blue token and one non-red token. We are not putting any blue tokens back in the bag, nor are we putting any green back in the bag, so this case will eventually run out as well, given none of the other cases are putting those colors back into the bag.

Case 2 does in fact put blue tokens into the bag, fueling case 3. The order of cases to cease executing are Case 2, because we will eventually run out of green tokens, as none are being put back into the bag in any of the other cases. Then Case 3, because after case 2 can no longer execute, blue tokens can no longer be put into the bag, and Case 3 will always remove at least one blue token. Then, when no more tokens can be put back into the bag, there will only be reds to pull, and do nothing as a result.

**Claim 1.** *Any combination of a bag of size  $n=2$  will terminate*

*Proof.* We can have the following combinations:

GG

Upon first pull, we follow case two. The bag now has GBB

Upon second pull, we can pull either

1. GB, put three red in. Bag now has BRRR. The third pull can be:
  - a. RR, do nothing. BR left. See BR (leads to terminate.)
  - b. BR, do nothing. RR left. See RR (leads to terminate.)
2. BB, put three red in. Bag now has GRRR.
  - a. RR, do nothing. GR left. See BR (leads to terminate.)
  - b. GR, do nothing. RR left. See RR (leads to terminate.)

No third pull - terminate.

GB

Upon first pull, we follow case three. The bag now has RRR. (see consequence of bag full of reds)  
terminate.

GR

Upon first pull, do nothing.  
terminate

BB

Upon first pull, we follow case three. The bag now has RRR. (see consequence of bag full of reds)  
terminate.

BR

Upon first pull, do nothing.  
terminate.

RR (or bag full of reds)

Each pull will result in do nothing, so pulling two reds is simply removing two reds from the bag and moving on the removing two more tokens. If the bag is full of reds, this is the only thing we will be doing until there are no reds left.  
terminate.

We can see that in any case, the bag will eventually become smaller than size 2 and the procedure will terminate.

□

**Claim 2.** *Any bag of size  $n$  will terminate.*

*Proof.* Any bag of size  $n$  will terminate, as the first pull will be one of the cases above. The only cases that result in more chips being put in are shown to eventually be drowned out by reds (only combinations available of color + red), or deplete in a color altogether. □

By the explanation (though poorly done with induction) we can see that cases two and three will eventually be unable to execute leaving only case one, which handles reds until they're all gone and the bag is empty (or has one element).