CS270: LAB #18

Gray Code

You may work in teams of ideally one or two people (three is acceptable in the event of an unscheduled absence). Unless stated otherwise, the lab is due to be submitted into Gradescope at the end of the class day (11:59pm)
In order to receive credit, follow these instructions:

[a] Every team member should be discussing simultaneously the same problem – do NOT try to divvy up the labor and assign different problems to different students since the material is cumulative.

[b] Directly edit this lab PDF using Sedja/PDFescape with your answers (extra pages can be added in the rare event you need more than the allotted space)

[c] Each lab, rotate which member has the responsibility of being the Scribe. This is the person that is typing the answers and uploading the final PDF – note that only a single copy of the filled in PDF is turned into Gradescope. Only one lab needs to be submitted for the entire team, and all members receive the same score. Make sure to use a font that your PDF editor is compatible with (otherwise you might find your answers appear as weird shapes/sizes or simply disappear entirely!)

[d] The Gradescope submission must have each answer properly tagged with the appropriate question. Moreover, every member of the team must be listed as a submitter. Although it is the Scribe which executes these actions, it is still the responsibility of the entire team to make certain this is done properly (thus it is highly recommended that the Scribe share their screen so the entire team can witness it). Answers which are improperly tagged cannot be seen by the grader and thus cannot be scored.

[e] **FOR REMOTE ONLY**: Each lab, rotate which member has the responsibility of being the Recorder. This is the person who hits the Zoom Record button (once the technical permission is granted by the TA/RCF/Professor) and ensures that everyone has their camera/microphone on. They are also the member that is responsible to make sure the DrexelStream video is marked as viewable and entered into the https://tinyurl.com/VidLinkForm webform before 11:59pm (they should also email the rest of their team as confirmation.) Note that the video file doesn't get created/processed until after the Recorder has quit Zoom.

[f] Each lab, rotate which member has the responsibility of being the Manager. This is the person that ensures that everyone is participating equally and honestly, keeps the group on task, ensures that all team members understand a solution before going on to the next question, and presses the "hand up" button in Zoom to summon a TA or the professor (but they only do so after surveying the group to make sure everyone has the same question).

Team Name (CS pioneer): Marvin Minsky

Scribe name: Lixiao Yang

Recorder name: Jerry Li

Manager name: Terie Ha

Other team member (if any): Evelyn Thai

Question 1: 8 points

Gray Codes are an important method for representing sequential values.

Read the following description of Gray Codes and answer the following questions.

`https://www.allaboutcircuits.com/technical-articles/gray-code-basics/`

(a) (1 point) What researcher at Bell Labs submitted the patent Pulse Code Communication?

Frank Gray

(b) (1 points) Are Gray Codes well suited for mathematical operations?

No

(c) (1 points) How many bits of the Gray Code sequence change as the number count progresses?

One bit

(d) (1 points) How can summing the bits of a Gray Code Sequence be used for error detection?

If the bits in a number are summed, the sum of the next number should only change by one with the sum alternating even and odd.

(e) (1 points) What was the problem cased by the standard binary system going from 7 to 8?

7 to 8 (a 1 unit increment) required all the bits representing the number to change, but with mechanical based systems that may not be the case depending on the individual mechanical responses and the timing of the read cycles.

(f) (1 points) Was the problem with mechanical or digit systems changing the bits?

Mechanical system

(g) (1 points) Why does a Gray Code reduce the chances of an error when going from 7 to 8?

Gray code only changes one of the bits for each transition, the chance for an error is reduced.

(h) (1 points) In the Gray Code image, what do the red and white boxes represent?

Each position on the disk corresponds to a binary sequence, with only one bit changing. Ordinary binary is in white boxes while Gray Code is in red.

(i) (1 points) How are Gray Codes used in modern aircraft?

The Gray Codes reflects a one bit change for each 100 foot increment allowing the altitude to be tracked.

Question 2 : 6 points

The smallest Gray Code is 1-bit.

$$G_1 = [0, 1]$$

Any larger Gray Code can be created using a recursive formula.

$$G_n = [0G_{n-1}, 1G'_{n-1}]$$

To create G , the following algorithm is followed:

1. Add a 0 to the beginning of all values in the previous Gray Code.
   $0G_{2-1} = 0G_1 = [00, 01]$
2. Reverse the previous Gray Code
   $G'_{2-1} = G'_1 = \text{reverse}([0, 1]) = [1, 0]$
3. Add a 1 to the beginning of the reversed code.
   $1G'_{2-1} = 1G'_1 = [11, 10]$
4. Append the two sequences created together.
   $[0G_{2-1}, 1G'_{2-1}] = [0G_1, 1G'_1] = [00, 01] + [11, 10] = [00, 01, 11, 10]$

Now let's create G_3 by performing those four steps from above.

(a) (1 point) Create $0G_2$

[000, 001, 011, 010]

(b) (1 point) Create G'_2 (Just reverse the elements of the list, not the individual bits)

[10, 11, 01, 00]

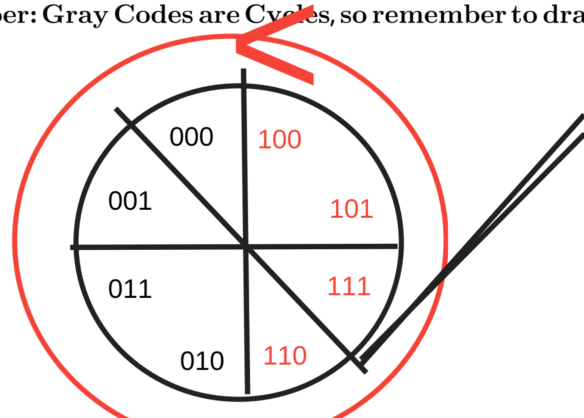
(c) (1 point) Create $1G'_2$

[110, 111, 101, 100]

(d) (1 point) Create G_3

[000, 001, 011, 010, 110, 111, 101, 100]

(e) (2 points) Verify you have created a Gray Code. Draw an arrow between each pair of binary values to show which bit changes. (if you do not have a drawing tool in your pdf editor, you can use color coding) **Remember: Gray Codes are Cycles, so remember to draw a line from the last element to the first.**



Question 3 : 25 points

(a) (15 points) Develop a Racket Function that takes a value and adds it to the front of every element in a list. Include a screenshot of your implementation. Give both a recursive version and non-recursive version

```
(define (prepend x L) ...)
```

Make sure your code passes all the below tests.

```
(equal? (prepend 0 '())
        '())
(equal? (prepend 0 '((0) (1)))
        '((0 0) (0 1)))
(equal? (prepend 0 '((0 0) (0 1) (1 1) (1 0)))
        '((0 0 0) (0 0 1) (0 1 1) (0 1 0)))
(equal? (prepend 1 '((0) (1)))
        '((1 0) (1 1)))
(equal? (prepend 1 '((0 0) (0 1) (1 1) (1 0)))
        '((1 0 0) (1 0 1) (1 1 1) (1 1 0)))
```

```
(define (prepend-recursive x L)
  (if (null? L)
      '()
      (cons (cons x (first L))
            (prepend-recursive x (rest L))
      )))

(define (prepend-non-recursive x L)
  (map (lambda (y) (append (list x) y)) L))
```

(b) (10 points) Develop a Racket Function to make Gray Codes and include a screenshot below

```
(define (gray_code n) ...)
```

Make sure your code passes all the below tests.

```
(equal? (gray_code 1)
        '((0) (1)) )
(equal? (gray_code 2)
        '((0 0) (0 1) (1 1) (1 0)))
(equal? (gray_code 3)
        '((0 0 0) (0 0 1) (0 1 1) (0 1 0)
         (1 1 0) (1 1 1) (1 0 1) (1 0 0)))
```

```
(define (gray_code n)
  (if (= n 1)
      '((0) (1))
      (append
       (prepend-recursive 0 (gray_code (- n 1)))
       (prepend-recursive 1 (my-reverse (gray_code (- n 1)))))))

(define (my-reverse L)
  (if (null? L)
      '()
      (append (my-reverse (rest L)) (list (first L)))))
```

Question 4 : 10 points

Two steps in creating a Gray Code require prepending a value to a Gray Code.

Assume that $G_n = [g_0, g_1, \cdots, g_x]$ is a Gray Code.

The first step to making G_n is prepending a 0 to each value.

$$0G_n = [0g_0, 0g_1, \cdots, 0g_x]$$

(a) (2 points) Explain (in plain English) why only one bits changes between $0g_y$ and $0g_{y+1}$ for all $0 \leq y < n$.

When the Gray Code starts with 0, the rest of the code is in binary sequence which only contains one bit change, when the Gray Code starts with 1, the mirrored sequence of the previous binary representation is still a one-bit change.

(b) (2 points) A Gray Code must cycle. Explain why only one bit changes between $0g_x$ and $0g_0$.

Since the sequence of the rest of the code is "mirrored", so the gx = g0, the only difference is the first digit, the first number starts with 0 and the last number starts with 1

(c) (2 points) Do the same arguments hold true if a 1 is prepended instead of 0? Why?

Yes, since that would be the reverse cycle of the Gray Code encoding wheel.

(d) (4 points) **Theorem 1**:

Prepending the same bit to a Gray Code keeps the property that only one bit changes between elements.

Question 5 : 12 points

Two steps in creating a Gray Code requires reversing all elements in a Gray Code.

Assume that $G_n = [g_0, g_1, \cdots, g_x]$ is a Gray Code.

$$G'_n = [g_x, g_{x-1}, \cdots, g_0]$$

(a) (4 points) Explain (in plain English) why only one bits changes between g_y and g_{y-1} for all $0 < y \leq n$.

Reversing the sequence doesn't change the relationship between adjacent elements.
If two elements differed by only one bit in the original sequence, they will continue to differ by only one bit when their positions are reversed.

(b) (4 points) A Gray Code must cycle. Explain why only one bit changes between the first and last element.

By the principle of the Gray Codes "mirrored" sequence, the first element 0g_0 and the last element 1g_0 have the same numbers exclude the first digit, which satisfies that only one bit changes between the first and last element.

(c) (4 points) **Theorem 2**:

__Reversing__ a Gray Code keeps the property that only __one bit__ changes between elements.

Question 6: 39 points
Prove by Induction that the following recursive formula creates a Gray Code.

To be a Gray Code, the list must satisfy each of the four requirements in the red box here:
You may use Theorems from previous questions.

The 4 requirements to be a GrayCode:

[a] each member is n bits [i.e. 0s and 1s]

[b] there are exactly 2^n members

[c] each value 0 to 2^n- 1 is represented exactly once (no repeats, no duplicates)

[d] exactly one bit changes from one member to the next (including the wrap around from the last back to the first)

$$G_1 = [0, 1]$$
$$G_n = [0G_{n-1}, 1G'_{n-1}]$$

(a) (3 points) Base Case: Explain why G_1 is a Gray Code

Each member is 1 bit, have exactly 2 members, each value 0 to 1 is represented exactly once, exactly one bit changes from the one member to the next.

(b) (4 points) **Inductive Hypothesis:** What should be assume about $G_n = [g_1, g_2, \cdots, g_x]$?

Assume that Gn = [g1, g2, ..., gx] is a Gray Code that satisfies all four requirements for some n.

(c) (8 points) We need to append two lists together to make $G_{n+1} = [0G_x, 1G'_x]$.
Explain why the last element in $0G_{n-1}$ only differs by 1 bit from the first element in $1G'_{n-1}$

The last element of 0G_n-1 would be the last element of G_n-1 but with a '0' prepend
The first element of 1G'_n-1 is just the first element of G'_n-1 but with a '1' prepend since it's the reverse of G_n-1
According to Theorem 2, the two new elements will differ at the prepended bit ('0' and '1') while keeping the remaining bits are the same.

(d) (8 points) Explain why the first element in $0G_{n-1}$ only differs by 1 bit from the last in $1G'_{n-1}$

The first element of 0G_n-1 is by prepending a '0' to the first element of G_n-1
The last element of 1G'_n-1 is formed by prepending a '1' to the last element of G_n-1
According to Theorem 1, the two elements will differ by only one bit.

(e) (8 points) Explain why G_{n+1} must have 2^{n+1} elements in it.

By the IH, we assume that G_n has 2^n elements. To form G_n+1, we append 0G_n and 1G'_n
Both 0G_n and 1G'_n will have 2^n elements each with only '0' or '1' prepended to each element of G_n
So G_n+1 will have 2^n + 2^n = 2^n+1 dlements

(f) (8 points) Explain why $G_{n+1} = [0G_n, 1G'_n]$ fulfills are four requirement to be a Gray Code (see Red box above).

Requirement [a]:

By appending a '0' to each member of G_n and a '1' to each member of G'_n, we effectively increase the length of each member by one bit. If each member of G_n is n bits (by IH), then each member of G_(n+1) will be n+1 bits.

Requirement [b]:
G_(n+1) is formed by appending 0G_n and 1G'_n. If G_n has 2^n members (by the inductive hypothesis), then G_(n+1) will have 2^n + 2^n = 2^(n+1) members.

Requirement [c]:
G_n by assumption covers all n-bit values exactly once. When we prepend '0' and '1' to form G_(n+1), we are essentially creating unique n+1-bit binary numbers. The '0' prepended numbers will cover the lower half of the n+1-bit range, and the '1' prepended numbers will cover the upper half. Together, they will cover all n+1-bit numbers exactly once.

Requirement [d]:
If exactly one bit changes between members of G_n (by the IH), then the same will be true for 0G_n and 1G'_n, since we're just adding a constant bit at the beginning of each member. Further, the last member of 0G_n and the first member of 1G'_n will differ by only the prepended bit ('0' vs '1'), thus maintaining the Gray code property even at the wrap-around point.

By satisfying all these four requirements, G_{n+1} = [0G_n, 1G'_n] is a valid Gray Code.