## Math 15 – Discrete Structures – §4.5 – Homework 16 Solutions

**<u>4.5#20</u>**: License plates in India begin with a code indentifying the state and district where the vehicle is registered, followed by a four-digit identification number. Once the sequence reaches 9999, a leter from the set $\{A, B, \ldots, Z\}$ is added (in order), and once these run out, additional letters are added and so on. So the sequence of identification numbers proceeds as follows: 0000, 0001, …,9999,A0000,A0001,…,A9999,B0000,B0001,…,B9999,….,Z0000,Z0001,…,Z9999,AA0000,AA0001,…

Write an algorithm in pseudo code that will print out the first 500,000 license plates for a given district in India.

ANS: We want to print the 10000 which are just numbers followed by the next 260000 which are a letter followed by 4 numbers, that will give us 270000, so we need another 230000 – and since the 23 letter of the alphabet is W, we can get these by preceeding A0000, A0001, …, W9999 by an "A". The code should look like this:

Let $A = \{A, B, \ldots, Z\}$ and let $N = \{0,1,2,3,4,5,6,7,8,9\}$.

```
for w ∈ N do
  for x ∈ N do
     for y ∈ N do
        for z ∈ N do
          print wxyz
for a ∈ A do
  for w ∈ N do
    for x ∈ N do
      for y ∈ N do
        for z ∈ N do
          print awxyz

for a ∈ {A,B,…W} do
  for w ∈ N do
    for x ∈ N do
      for y ∈ N do
        for z ∈ N do
           print Aawxyz
```

**<u>4.5#24</u>**: Let $x_1, x_2, \ldots, x_n$ be an array. Consider the following algorithm.

```
for i ∈ {1, 2, ... , ⌊n/2⌋} do
   ┌ t ← x_i
     x_i ← x_{n−i+1}
   └ x_{n−i+1} ← t
```

a. How many ⟵ operations does this algorithm perform? Your answer should be a function of *n*.
   SOLN: $3 \cdot \left\lfloor \frac{n}{2} \right\rfloor$
b. What does this algorithm do to the array?
   SOLN: It continually swaps the *i*th element with the $n + 1 - i$th element starting with the first and continuing up until, but not including the middle, effectively reversing the array.

**<u>4.5#26</u>**: Let $x_1, x_2, \ldots, x_n$ be an array of integers. Write a pseudocode algorithm that will compute the probability that a randomly chosen element of this array will be odd.

SOLN:

```
c ← 0
for i ∈ {1,2, … , n) do
   if x_i mod2=1 then
   c ← c + 1
print c/n
```

**4.5#28**: Let Write a pseudocode algorithm that will print out all strings of four symbols from the set $A = \{A, B, \dots, Z\}$ such that no symbol is repeated. How many such strings are there?

ANS: There are $26 \cdot 25 \cdot 24 \cdot 23 = 358800$ permutations of 4 symbols chosen from a set of 26.

```
for w ∈ A do
  for x ∈ A\{w} do
    for y ∈ A\{w,x} do
      for z ∈ A\{w,x,y} do
        print wxyz
```

**4.5#29**: Write a pseudocode algorithm that prints out all allowable colorings of the vertices $a, b, c,$ and $d$ of a graph in the shape of a quadrilateral as a four-symbol string using the symbols in $C = \{R, G, B, V\}$. Use the two disjoint cases: when *b* and *d* are the same color, and when *b* and *d* are different colors. A coloring where adjacent vertices are the same color is not allowed.

SOLN: Let $C = \{R, G, B, V\}$

```
for a ∈ C do  #3  or 4 colors
  for b ∈ C\{a} do
    for d ∈ C\{a,b} do
      for c ∈ C\{b,d} do
        print abcd
for a ∈ C do #2 or 3 colors
  for b ∈ C\{a} dp
    for c ∈ C\{b} do
      print abcb
```

If you trace the first one you get

| *a* | *b* | *c* | *d* |
|---|---|---|---|
| R | G | R | B |
| R | G | V | B |
| R | G | B | V |
| R | G | R | V |
| R | B | R | G |
| R | B | V | G |
| R | B | R | V |
| R | B | G | V |
| R | V | R | G |
| R | V | B | G |
| R | V | R | B |
| R | V | G | B |

This pattern will continue like this for *a* = G, B and V, producing a total of 48 different colorings. For each of the $4 \cdot 3 \cdot 2 = 24$ choices for *a*, *b* and *d*, there are 2 choices for *c*.

Note that the *abcd* pattern where $b \neq d$ will always produce different outcomes than the *abcb* pattern, whose sequences are produced by the second algorithm.

For the second, the trace looks like this:

| *a* | *b* | *c* | *d* |
|---|---|---|---|
| R | G | R | G |
| R | G | B | G |
| R | G | V | G |
| R | B | R | B |
| R | B | G | B |
| R | B | V | B |

…and so on…

For each of the $4 \cdot 3 = 12$ choices for *a* and *b* there are 3 choices for *c*, so there are 36 of these type.

**4.5#30**: Repeat the previous exercise using two different colors, three different colors and four different colors.

SOLN: Let $C = \{R, G, B, V\}$

```
for a ∈ C do  #only two colors
  for b ∈ C\{a} do
    print abab
for a ∈ C do #3 colors
  for b ∈ C\{a} dp
    for d ∈ C\{a,b} do
      print abad
for a ∈ C do  #these are isomorphic to the previous, depending…
  for b ∈ C\{a} do
    for d ∈ C\{a,b} do
      print bada
for a ∈ C do   #4 colors
  for b ∈ C\{a} do
    for d ∈ C\{a,b} do
      for e ∈ C\{a,b,d} do
        print abde
```