DISCRETE MATH 37181 TUTORIAL WORKSHEET 7

INSTRUCTIONS. Complete these problems in groups of 3-4 at the whiteboard. Partial solutions at the end of the PDF.

Notation: a *binary string* of length $n \in \mathbb{N}_+$ is an expression of the form $d_1 d_2 \dots d_n$ where $d_i \in \{0, 1\}$ for $1 \leqslant i \leqslant n$. For example, 11011 is a binary string of length 5. The set of all binary strings of length n is denoted $\{0, 1\}^n$.

Whenever people talk about strings of finite length, it is helpful to include strings of length 0. For this subject, let us denote the (unique) string of length 0 by the symbol λ.

If a, b, u, v are strings, we call u a *factor* of v if $v = aub$.

1. (a) How many binary strings of length 4 do not contain a factor 11?
   (b) How many binary strings of length n do not contain a factor 11?
   (c) How many binary strings of length n do not contain a factor 11 and have final digit 1?

2. Define a relation $\mathscr{T}$ on the set of all finite length binary strings by $a \mathscr{T} b$ if

   – the sum of the digits in a is strictly less than the sum of the digits in b, or
   – a is obtained from b by deleting one digit from b.

   Which of the following is true?

   A. $\mathscr{T}$ is reflexive
   B. $\mathscr{T}$ is antisymmetric
   C. $\mathscr{T}$ is transitive
   D. $0110 \mathscr{T} 11$
   E. $11 \mathscr{T} 0110$
   F. none of (A)–(E).

3. A *boolean function* is a function of the form $f : \{0, 1\}^n \to \{0, 1\}$.
   What is the number of boolean functions of the form $f : \{0, 1\}^n \to \{0, 1\}$?

4. In the lectures Murray gave a "combinatorial proof" of the binomial theorem (formula to expand out $(x + y)^n$). Maybe you thought it was dodgy. Give a proof by induction instead. Hint you may need to use this lemma (and prove it first):

$$\binom{n}{i-1} + \binom{n}{i} = \binom{n+1}{i}$$

5. Use the Binomial theorem to prove the following identities: [1]
   (a) $\binom{n}{0} - \binom{n}{1} + \binom{n}{2} - \cdots + (-1)^n \binom{n}{n} = 0.$
   (b) $\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n} = 2^n$

Date: Week 7 workshop (Wednesday 6, Thursday 7, Friday 8 April).

[1]Hint: easy, just plug into the formula. No proofs required since the theorem is already proved so you can just use it.

6. (a) Give a combinatorial proof [2] of the following fact

$$\binom{n}{i-1} + \binom{n}{i} = \binom{n+1}{i}$$

[3]

(b) Draw Pascal's triangle (to compute binomial coefficients without needing a calculator) using the above fact, down to 10 rows.

$$\begin{array}{ccccc} & & \binom{0}{0} & & \\ & \binom{1}{0} & & \binom{1}{1} & \\ \binom{2}{0} & & \binom{2}{1} & & \binom{2}{2} \end{array} \qquad \begin{array}{ccccc} & & 1 & & \\ & 1 & & 1 & \\ 1 & & 2 & & 1 \end{array}$$

7. Consider strings consisting of the symbols '(' and ')'. Such a string is *balanced* if every '(' has a unique matching ')' to its right[4]. For example (()) and ()(()()) are balanced but (()))(() is not.

Let c_n be the number of balanced bracket strings of length $2n$.

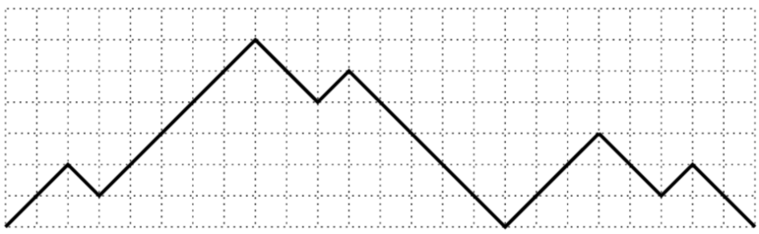
(a) Compute c_0, c_1, c_2, c_3.

(b) Prove that $c_{n+1} = \sum_{i=0}^{n} c_i c_{n-i}$ [5]

(c) Use the recursive formula to compute c_4, c_5, c_6.

[6]

8. A *Dyck-path* in the xy-plane is a path consisting of $2n$ diagonal lines of length $\sqrt{2}$ and slope ± 1, starting at $(0,0)$ and ending at $(0,2n)$, and never going below the x-axis.



How many different Dyck-paths from $(0,0)$ to $(0,2n)$ are there?

ENJOY YOUR STUVAC BREAK!

[2]combinatorial proof means the LHS and the RHS are two different ways to count the same object. In this case the object is the number of subsets of size i of a set of size $n+1$.

[3]Hint: you might want to use this in Question 3. Or you proved it using the ! formula.

[4]More formally, we can define 'being balanced' recursively by saying the empty string is balanced, and if u, v are balanced then $(u)v$ is balanced.

[5]Hint: consider the first time the prefix of the string becomes balanced.

[6]XHMTL requires you to write balanced code

Brief solutions:

1. (b) Recursive. If b_n is the number of binary strings of length n without a 11 factor, then $b_0 = 1$ and $b_1 = 2$. A string of length $n \geqslant$ either starts with 0 or 1. If it starts with 0, the next letter can be anything, so there are b_{n-1} possible strings. If it starts with 1, the next letter must be 0, and then anything, so there are b_{n-2} possible strings. So in total $b_n = b_{n-1} + b_{n-2}$. This is the Fibonacci sequence (starting at $1, 2$).

   (c) If c_n is the number of strings without 11 and ending with 1, then $c_0 = 0, c_1 = 1$. For $n \geqslant 2$ the last two digits must be 01 and we have b_{n-2} possible prefixes. So this is also the Fibonacci sequence (starting at $0, 1$).

3. For each boolean string $\vec{b}$ of length n (that is, $\vec{b}$ is an element in $\{0,1\}^n$), there are two possible values for f on $\vec{b}$. There are 2^n boolean strings of length n. So the number is
$$2^{2^n}.$$

7. (a) $c_0 = 1$ (there is one way to write nothing), $c_1 = 1$: $()$, $c_2 = 2$: $()(), (())$, $c_3 = 5$: $()()(), ()(()), (())(), (()()), ((()))$.

   (b) For each string of length $2n+2$, there is always a "first time" the prefix is balanced. For example $((())())()$ the first time is after 8 brackets. The balanced prefix always looks like (w) where w is a balanced string itself (possibly empty): if not then you could find a shorter prefix that is balanced.

   So the number of ways to arrange the balanced prefix of length $2i+2$ is c_i, and the ways to write the suffix is c_{n-i} since there are $2(n-i)$ brackets left, thus $c_i c_{n-i}$ total such strings.

   Every string has a unique i so that the first time the prefix is balanced is at $2i+2$ so we can add up all the cases for each i without even counting the same string twice. So $c_{n+1} = c_0 c_n + \cdots + c_n c_0$ as required.

   (c) $c_4 = 14, c_5 = 42, c_6 = 132, c_7 = 429, c_8 = 1430, c_9 = 4862, c_{10} = 16796$.

8. The set of paths to $(0, 2n+2)$ can be broken up into those that return to the x-axis for the *first time* after $2i$ steps.
   Answer: same as question 6: c_n.