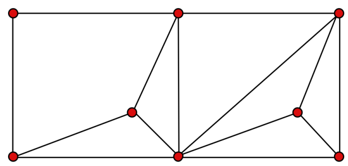
Math 232, Spring 2018, Boris Botvinnik

REVIEW FOR THE FINAL TEST, PART I:

(1) Determine the chromatic polynomial $P(G,\lambda)$ for the following graph. Find a number of proper colorings of the graph G with 5 colors.



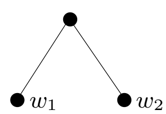
(2) An algebraic expression is written in the reverse Polish notations as follows:

$$- \ / \ - \ \wedge \ x \ 3 \ 1 \ - \ x \ 1 \ + \ \wedge \ x \ 2 \ + \ x \ 1$$

(a) Find a binary tree representing this algebraic expression.
(b) Find this algebraic expression.
(c) Write this expression in the Polish notations.

(3) Consider the Huffman Algorithm:

Huffman$(L = \{w_1, w_2, \dots, w_k\})$:
$\{$`Input: A list of weights:` $L = \{w_1, w_2, \dots, w_k\}$, $k \geq 2\}$
$\{$`Output: an optimal tree` $T(L)\}$
`if` $k = 2$ `then`
`return the tree`



`else`
`Choose two smallest weights` u `and` v `of` L .
`Make a list` L' `by removing the elements` u `and` v `and ading the element` $u+v$.
`Let` $T(L') :=$**Huffman**(L') .
`Form a tree` $T(L)$ `from` $T(L')$ `by replacing a leaf of weight` $u+v$
`by a subtree with two leaves of weights` u `and` v .

`return` $T(L)$.

Prove that the algorithm **Huffman**(L) does produce an optimal binary tree for the weights $L = \{w_1, w_2, \dots, w_k\}$.

(4) Find an Euler circuit (if it does exist) in a given graph.

(5) Describe the most effective way how to merge together the ordered lists L_1, L_2, L_3, L_4, L_5, L_6, L_7, L_8, L_9 with the lengths $|L_1| = 17$, $|L_2| = 12$, $|L_3| = 9$, $|L_4| = 30$, $|L_5| = 32$, $|L_6| = 44$, $|L_7| = 15$, $|L_8| = 10$, $|L_9| = 44$.

(6) Use generating functions to resolve the recurrence relation: $x_0 = 2$, $x_n = 3x_{n-1} - 4n$.

(7) Let $\Sigma = \{0,1\}$ and A_n be the set of binary strings of length n which do not contain the string 10. Find and solve a recurrence relation for $a_n = |A_n|$.

(8) Solve the following recurrence relations:
(a) $a_n = 5a_{n-1} + 6a_{n-2}$, $n \geq 2$,
$a_0 = 0$, $a_1 = 1$.
(b) $a_n = 2a_{n-1} - 2a_{n-2}$, $n \geq 2$,
$a_0 = 1$, $a_1 = 1$.

(9) Prove that if a finite graph $G = (V,E)$ in which each vertex has degree at least 2 contains a cycle.

(10) Prove that if a finite graph $G = (V,E)$ is a tree, if and only if $|V| = |E| + 1$.

(11) Let G_n be a graph which is obtained from the complete graph K_n by deleting one edge. Determine the chromatic polynomial $P(G_n, \lambda)$ and the chromatic number $\chi(G_n)$.

(12) Use generating functions to solve the following recurrence relations

(a) $a_n = a_{n-1} + n$ for $n \geq 1$, and $a_0 = 1$;

(b) $a_n = 5a_{n-1} - 6a_{n-2}$, $a_0 = 1$, $a_1 = -2$;

(c) $a_n = a_{n-1} + 8a_{n-2} - 12a_{n-3}$, $a_0 = 2$, $a_1 = 3$, $a_3 = 19$.

(d) $a_n = -3a_{n-1} + 10a_{n-2} + 3 \cdot 2^n$, $n \geq 2$, and $a_0 = 2$, $a_1 = 1$.

(13) Find and solve a recurrence relation for a_n, the number of 012-strings of length n in which no 2 is followed (immediately or after some intervening characters) by a 1.

(14) Let a_n be the number of words of lenght n in A, B, C, and D with an odd number of B's. Calculate a_0, a_1, a_2, a_3, a_4. Find a recurrence relation satisfied by a_n for all $n \geq 2$.

(15) Consider the Euclid Algorithm:

EUCLID (m, n):
{`Input:` $m, n \in \mathbf{N}$ `, not both` 0}
{`Output:` `gcd`(m, n)}
`if` $n = 0$ `then`
  `return` m
`else`
  `return` **EUCLID** $(n,\ m \text{ MOD } n)$

Use the algorithm to find gcd(21,231, 141).

(16) Let $G = (V, E)$ be a loop-free finite graph with $|V| = n$. Prove that G is a tree if and only if its chromatic polynomial $P(G, \lambda) = \lambda(\lambda - 1)^{n-1}$

(17) Let C_n be a cycle of length n. Prove that $P(C_n, \lambda) = (\lambda - 1)^n + (-1)^n(\lambda - 1)$.

(18) Let W_{n+1} be a "wheel" with $n+1$ vertices. Prove that $P(W_{n+1}, \lambda) = \lambda(\lambda-2)^n + (-1)^n\lambda(\lambda-2)$.

(19) Here is the algorithm **Tree**.

Tree (G, v)
`Input:` A vertex v of the finite graph G
`Output:` A set E of edges of a spanning tree for the component of G that contains v
`Let` $V := \{v\}$ `and` $E := \emptyset$
  `(where` V `is a list of visited vertices).`
`while there are edges of` G `joining vertices in` V `to vertices that are not in` V `do`
  `Choose such an edge` $\{u, w\}$ `with` $u \in V$ `and` $w \notin V$.
  `Put` w `in` V `and put the edge` $\{u, w\}$ `in` E.
`return` E

Prove that **Tree** (G, v) produces a spanning tree for the component of G containing the vertex v.

(20) Here is the Prim's Algorithm:

Prim's Algorithm $(G = (V(G), E(G)),\ \mathtt{wt} : E(G) \to (0, \infty))$
`Input:` A finite weighted connected graph $(G, \mathtt{wt})$ with edges listed in any order
`Output:` A set E of edges of an optimal spanning tree for G)
`Set` $E = \emptyset$. `Choose` w `in` $V(G)$ `and set` $V := \{w\}$.
`while` $|V| < |V(G)|$ `do`
  `Choose an edge` $\{u, v\}$ `in` $E(G)$ `of smallest possible weight`
    `with` $u \in V$ `and` $v \in V(G) \setminus V$.
  `Put` $\{u, v\}$ `in` E `and put` v `in` V.
`return` E

Prove that **Prim's** $(G = (V(G), E(G))$ produces an optimal spanning tree for G.