## 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:

$$
-/-\wedge x 31-x 1+\wedge x 2+x 1
$$

(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:
$\operatorname{Huffman}\left(L=\left\{w_{1}, w_{2}, \ldots, w_{k}\right\}\right)$ :
\{Input: A list of weights: $\left.L=\left\{w_{1}, w_{2}, \ldots, w_{k}\right\}, k \geq 2\right\}$
\{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 $L^{\prime}$ by removing the elements $u$ and $v$ and ading the element $u+$ $v$.

Let $T\left(L^{\prime}\right):=\mathbf{H u f f m a n}\left(L^{\prime}\right)$.
Form a tree $T(L)$ from $T\left(L^{\prime}\right)$ 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 $L=\left\{w_{1}, w_{2}, \ldots, w_{k}\right\}$.
(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 $\left|L_{1}\right|=17,\left|L_{2}\right|=12,\left|L_{3}\right|=9,\left|L_{4}\right|=30,\left|L_{5}\right|=32$, $\left|L_{6}\right|=44,\left|L_{7}\right|=15,\left|L_{8}\right|=10,\left|L_{9}\right|=44$.
(6) Use generating functions to resolve the recurrence relation: $x_{0}=2, x_{n}=3 x_{n-1}-4 n$.
(7) Let $\Sigma=\{0,1\}$ and $A_{n}$ be the set of binary strings of length $n$ which do not contain the string 001. Find and solve a recurrence relation for $a_{n}=\left|A_{n}\right|$.
(8) Solve the following recurrence relations:
(a) $a_{n}=5 a_{n-1}+6 a_{n-2}, n \geq 2$, $a_{0}=0, a_{1}=1$.
(b) $a_{n}=2 a_{n-1}-2 a_{n-2}, n \geq 2$, $a_{0}=1, a_{1}=1$.
(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 $P\left(G_{n}, \lambda\right)$ and the chromatic number $\chi\left(G_{n}\right)$.
(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) $a_{n}=5 a_{n-1}-6 a_{n-2}, a_{0}=1, a_{1}=-2$;
(c) $a_{n}=a_{n-1}+8 a_{n-2}-12 a_{n-3}, a_{0}=2, a_{1}=3, a_{3}=19$.
(d) $a_{n}=-3 a_{n 1}+10 a_{n 2}+3 \cdot 2^{n}, n \geq 2$, 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: \(\operatorname{gcd}(m, n)\}\)
if \(n=0\) then
    return \(m\)
else
return \(\operatorname{EUCLID}(n, m \operatorname{MOD} n)\)
```

Use the algorithm to find $\operatorname{gcd}(21,231,141)$.
(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 $P\left(C_{n}, \lambda\right)=(\lambda-1)^{n}+(-1)^{n}(\lambda-1)$.
(18) Let $W_{n+1}$ be a "wheel" with $n+1$ vertices. Prove that $P\left(W_{n+1}, \lambda\right)=\lambda(\lambda-2)^{n}+(-1)^{n} \lambda(\lambda-$ 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 $(G=(V(G), E(G))$, wt : $E(G) \rightarrow(0, \infty))$
Input: A finite weighted connected graph $(G, w t)$ 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 $v \in V(G) \backslash V$.
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 $G$.

