CMPUT 204 (A1)
Midterm 3: Version A
November 19, 2004

Department of Computing Science
University of Alberta

Instructor: M. Bowling

Time: 50 Minutes

Last Name:_____

First Name:_____

- Put your name above (no ID). Print clearly.
- There are 4 problems and 6 total pages (50 marks in total).
- Closed Book: Only Pen, Pencil & Eraser allowed.
- Use space below the questions to write your answers.

Do NOT write on this page, for instructor's use only.

total marks	question	your marks
10	1	
5	2	
20	3	
15	4	
50		

Last Name:_____ **ID#:**_____

---

1. **(10 Marks)** Dr. Strange has decided to invent his own sequence of numbers, which he calls the Multinacci sequence. The sequence is defined by the following recurrence.

$$M(n) = \begin{cases} 1 & \text{if } n = 1 \\ 1 & \text{if } n = 2 \\ M(n-1)M(n-2) + 1 & \text{otherwise} \end{cases}$$

Use dynamic programming to write the procedure $\textsc{Mult}(n)$ in pseudocode to compute the nth Multinacci number. The procedure must run in $O(n)$ time.

Solution: See the solution to question 3 on midterm 2.

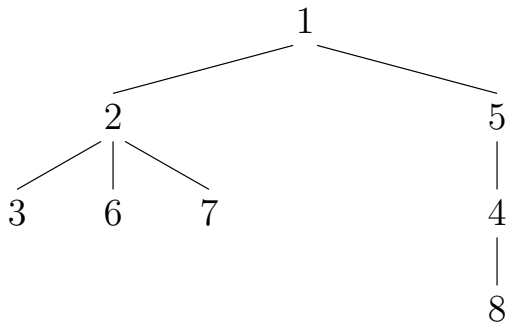
Last Name:_____ **ID#:**_____

---

2. **(5 Marks)** Draw the tree that results from a breadth-first traversal of the following simple undirected graph starting from vertex "1". Assume that adjacent edges are iterated through in ascending numerical order.

$$
\begin{aligned}
V &= \{1,2,3,4,5,6,7,8\} \\
E &= \{(1,2),(1,5),(2,3),(2,5),(2,6),(2,7), \\
&\quad (3,4),(3,5),(4,5),(4,8),(5,7),(6,7)\}
\end{aligned}
$$

Solution:



Last Name:_____ **ID#:**_____

---

3. **(20 Marks)** Consider a city and its public transportation system. The city has a set of locations L and a set of routes R. Each route is itself a subset of at most k locations, which the route stops along. In addition, the routes are partitioned into three classes "bronze" (R_b), "silver" (R_s), and "gold" (R_g). Three passes can be purchased: bronze allows unlimited travel on bronze routes, silver allows unlimited travel on silver and bronze routes, and gold allows unlimited travel on all routes. The passes, though, are increasingly more expensive.

Use a disjoint sets data structure (i.e., UNION-FIND) to write a procedure that given L, $R_{\{s,g,b\}}$, and two locations l_1 and l_2, returns the cheapest option that will allow travel between the two locations. The length or number of route changes is irrelevant.

Solution:

```
CheapestPass(L, R_g, R_s, R_b, l_1, l_2)
    for l ∈ L do
        MakeSet(l)
    for R ∈ R_b do
        for r ∈ R do
            Union(r, R[0]) // R[0] is some element in R
    if Find(l_1) == Find(l_2) then return b
    for R ∈ R_s do
        for r ∈ R do
            Union(r, R[0])
    if Find(l_1) == Find(l_2) then return s
    for R ∈ R_g do
        for r ∈ R do
            Union(r, R[0])
    if Find(l_1) == Find(l_2) then return g
    return nil
```

Last Name:_____ **ID#:**_____

---

4. **(15 Marks)** (a) A k-coloring of a graph is a labeling of the graph's vertices, $c[v] \in \{1, \ldots, k\}$ with the additional property that adjacent vertices cannot have the same labeling, i.e.,

$$u \in Adj[v] \Rightarrow c[u] \neq c[v].$$

Write a procedure that given a simple undirected graph $G = (V, E)$, k, and a coloring $c[\cdot]$, returns true if and only if it is a k-coloring.

Solution:

```
CheckColoring(V, E, c, k)
    for v ∈ V do
        if c[v] ∉ {1, ..., k} then return false
        for u ∈ Adj[v] do
            if c[u] = c[v] then return false
```

BONUS(2.5 marks): What is the running time of your procedure in terms of $|V|$ and $|E|$ if G is represetend with adjacency lists?

Solution: $\Theta(|V| + |E|)$

BONUS(2.5 marks): What is the running time of your procedure in terms of $|V|$ and $|E|$ if G is represetend with an adjacency matrix?

Solution: $\Theta(|V|^2)$