CMPUT 204 Winter 2001: Section B2

Midterm ~~1~~ 2

Monday, Mar. 12

Time: 50 minutes

Weight 20%

Total Points 40

Last name:
First name:
Unix ID:

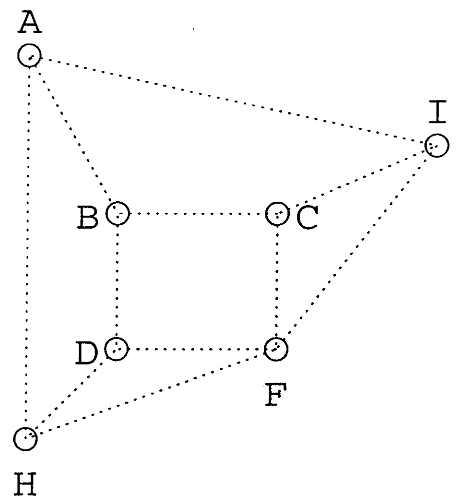
- An $8\frac{1}{2} \times 11$ formula sheet is allowed.
- No books or other notes are allowed.
- No calculators or other mechanical devices are allowed.
- The term "graph" in this exam always refers to undirected graphs.
- This exam has 4 pages and 5 questions. You are responsible for checking that your exam booklet is complete.

Question 1 [10 points]

Consider the following graph traversal algorithm. Note that it identical to BFS, except that it uses a stack instead of a queue.

```
void search(Graph G, Vertex s, Color color[]){
   create color[] and initialize to white
   Stack pending = new Stack();
   color[s] = gray;
   pending.push(s);
   while (!pending.empty()) {
      v = pending.pop();
      for each Vertex w in Nbd(v) { // using adjacency list order
         if (color[w]==white) {
            color[w] = gray;
            pending.push(w);
            // process tree edge vw
         } else {
            // process nontree edge vw
         }
         // process vertex v
      }
      color[v] = black;
   }
```

1.a [4 pts]: Draw the search tree given by this algorithm applied to the following graph. Use `A` as the starting vertex, and write numbers on the edges (both tree and non-tree) to indicate the order in which they are processed.




```
A: H,B,I
B: C,D
C: F,I,B
D: B,H,F
F: I,H,C,D
H: F,D,A
I: A,C,F
```

1.b [3 pts]: Can the non-tree edges be cross edges or back edges or both? Briefly justify why your answer holds in general.

1.c [3 pts]: How does the depth of the tree compare to those given by BFS and DFS? Briefly justify why your answer holds in general.

Question 2 [3 points]

Suppose we were to modify Strassen's algorithm so that instead of dividing each matrix into four $\frac{n}{2} \times \frac{n}{2}$ pieces, we divided them into nine $\frac{n}{3} \times \frac{n}{3}$ pieces. If 26 recursive calls were needed, then what would the asymptotic complexity of this algorithm be? (Note: $log_3 26 \approx 2.97$, $\lg 26 \approx 4.70$, $\lg 3 \approx 1.58$, but you may leave any logarithms in your answer unevaluated)

Question 3 [10 points]

For each of the following functions write expressions describing the number of multiplications done. Do not try to solve the expressions.

3.a [5 pts]:

```
int foo(int n) {
   int i, j, k, val;
   int val = 0;
   for (i=0; i<n; i++)
      for (j=0; j<i; i++)
         for (k=j; k<n; i++)
            val += i+j*k;
   return val;
}
```

3.b [5 pts]:

```
int bar(int n) {
   int val;
   if (n==0) val = 1;
   else if (n==1) val = 2;
   else val = bar(n-2) * bar(n/2);
   return val;
}
```

Question 4 [10 points]

Recall the formula

$$\sum_{v \in V} d(v) = 2 \cdot |E|$$

which holds for any graph $G = (V, E)$, where $d(v)$ is the degree of the vertex v.

4.a [5 pts]: Briefly explain why this formula is true.

4.b [5 pts]: What is the relevance of this formula?

Question 5 [12 points]

5.a [10 pts]: Adapt depth-first search to determine whether there is an odd length cycle in a connected component of a graph.

preconditions:
- the inputs are graph $G = (V, E)$ and starting vertex $v \in V$
- G has n vertices, m edges
- the set of edges is represented by an adjacency list

postconditions:
- The algorithm returns a boolean value odd which is true iff an odd cycle exists in the connected component containing v.

5.b [2 pts]: What is the complexity of your algorithm in part **a.**?

Q	Mark	
1		10
2		3
3		10
4		10
5		12
Σ		40