

CS 61B SUMMER 1995

Midterm#2

Instructor: Unknown

Problem #1

Your goal is to determine what the following program prints. You may wish to fill in the comments in order to aid in partial credit, but you will get full credit if the answer is correct.

```
#include < iostream.h >

int c1=0, c2=0, c3=0, c4=0, c5=0;

class David {
public:
    David() {c1++;}
    David(David &dave) {c2++;}
    David(int mary) {c3++;}
    ~David() {c4++;}
    David &operator=(const David & dave) {c5++;}
};

void f(David a){}
void g(David & a){}

main() {
    David a, b;
    {
        David c(a);
        David d(3);
        David *e = new David;
        f(d);
        g(d);
    }
    Count << "Answer: " << c1 << c2 << c3 << c4 << c5 << endl;
}
```

Fill in the answer below.

Answer:

Problem #2

For each of the following assertions, indicate if it is TRUE or FALSE.

$n = O(n)$	
$n + n \log n = O(n^2)$	
$n + n^3 = O(n \log n)$	
$n^3 = \Theta(5n^3 + 3n^2)$	
$n + \log n = \Theta(n)$	
$n + \log n = \Theta(n \log n)$	

Problem #3

Prove by induction that the sum of the first n odd integers is n^2 . In other words, show, for $i = 1 \rightarrow n$, $\text{Sigma}(2i - 1) = n^2$

Problem #4

An undirected graph is stored as an adjacency matrix or as an adjacency list as below. naturally, I only listed those portions of the class definition you'll need for the question.

```
typedef int Vertex; // "Vertex" means the same as "node."

class AdjacencyMatrix {
  typedef Array Column
  public:
    AdjacencyMatrix (const AdjacencyList &);
    Boolean operator() (Vertex u, Vertex v) {return graph[u][v];}
  private:
    Array<column> graph;
};
```

```

class AdjacencyList {
    typedef List ListOfAdjacentVertices;
public:
    //List of vertices connected to vertex u
    List neighbors (Vertex u) {return neighborLists[u];}

    //Number of vertices in the graph
    int vertexCount() {return neighborLists.getSize();}
private:
    Array neighborLists;
};

class ListIterator { //The portions you need ...
    T operator() ();
    void start(const List & l);
    void operator++();
    Boolean done();
};

```

Write the following constructor function, which converts an adjacency list representation of a graph to an adjacency matrix. You may not use any member functions of class `List`. You should use the `ListIterator` for efficiency.

```

AdjacencyMatrix (const AdjacencyList &g) {
    //Converts graph g to an adjacency matrix representation
    //About 6-10 lines of code.

}

```

Problem #5

After taking this class, Alice knows that quicksort is typically faster than bubblesort.

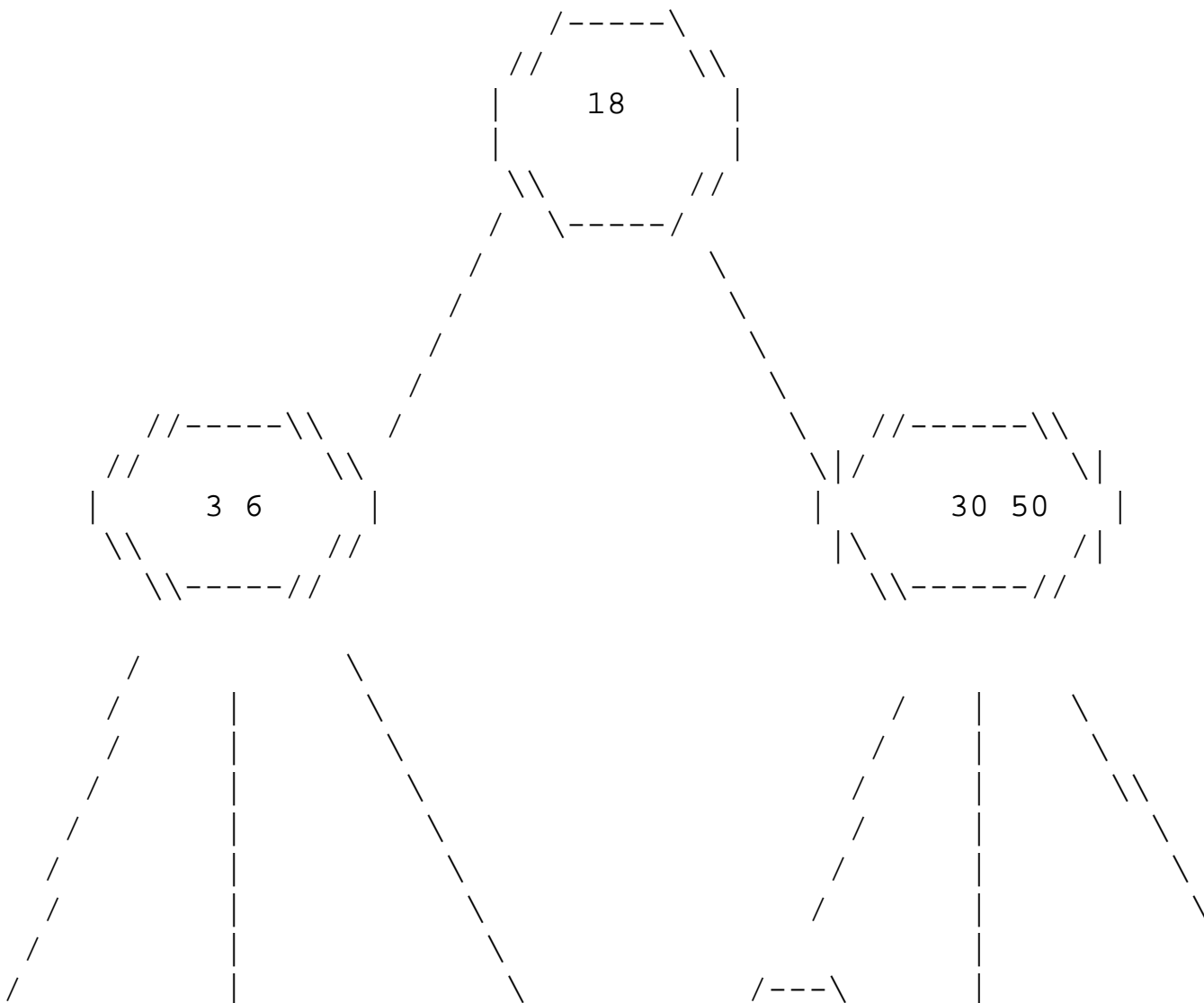
(a) (5 points) What theoretical evidence does she have of this? (One sentence).

(b) (10 points) She runs bubblesort and randomized quicksort on the same data, and even though both were coded properly and efficiently in c++, bubblesort ran faster. Give two reasons why this might be the case.

- Reason 1:
- Reason 2:

Problem #6

Draw the resulting 2-3 tree after the key 32 is inserted:



1		8	20	//--\\	70
	5		\-24	35 38	
				\\--//	

**Posted by HKN (Electrical Engineering and Computer Science Honor Society)
University of California at Berkeley**

**If you have any questions about these online exams
please contact <mailto:examfile@hkn.eecs.berkeley.edu>**