

COP 3530
Data Structures

Midsemester Exam

Name: _____

Email: _____

June 15, 1998

This exam has 4 questions. Each question starts on a new page. Each page is worth 50 points. Please answer each question on its page. You may write on the back of a page.

1. This question refers to the function `CanSum` defined below. `CanSum` returns `true` if there are two integers in the array `A` that sum to exactly `K`. Ignore all syntax errors.

```
bool CanSum( const Vector<int> & A, int K )
{
    for( int i = 0; i < A.Length( ); i++ )
        for( int j = i + 1; j < A.Length( ); j++ )
            if( A[ i ] + A[ j ] == K )
                return true;

    return false;
}
```

- (a) What is the Big-Oh running time of this function?
- (b) If it takes 10 seconds to run `CanSum` on an array of 10000 elements, approximately how long will it take to run `CanSum` on an array of 40000 elements (assuming that low-order terms are negligible)?
- (c) Describe a more efficient algorithm than the one above and give its running time. You may not use any code in your description.

2. Write a complete generic queue class. You must support functions named `IsEmpty`, `MakeEmpty`, `Enqueue`, `Dequeue`, and `GetFront`. Your class should contain one data member of type `list<Object>` (`list` is the STL class), and no other data members. Make sure you signal errors.

3. (a) Write a function `NumWithOneChild`, started below. `NumWithOneChild` is passed a pointer to the root node of a binary tree and returns the number of nodes in the tree that contain exactly one child. You may not add any global or static variables, or alter the function signature.
- (b) What is the running time of your algorithm?

```
int NumWithOneChild( Node *T )  
{
```

4. Describe in English an algorithm to solve the following problem, and give its Big-Oh running time. Code is not necessary, and is in fact discouraged.

*Let array A contain N **positive** numbers. Determine the maximum value of $A[j]/A[i]$, where $j \geq i$. For instance, if the array contains 4, 6, 5, 2, 4, 5, 1, 3, then the maximum value is 3, determined by the seventh and eighth elements.*

I will accept any subquadratic solution. A quadratic solution is trivial and gets no credit. *Hint: This problem is similar to the maximum sequence sum problem.*