

Open book and notes. Show all of your work clearly in the space provided or on the additional page at the end of the exam. If the additional page is used, be sure to clearly label the content for each problem. Be sure to *read each problem carefully*. You should answer all 6 questions, and if time permits, you may wish to answer the bonus question as well. Note that the exam is double sided.

1. (20 points) $f(n)$ describes the actual time consumed by an algorithm on an input of size n . Give the corresponding big-oh notation for $f(n)$ in each part. (In order to receive partial credit, you must explain your reasoning.)

(a)

$$f(n) = n^2 + \log n \quad \text{for all } n > 0$$

(b)

$$f(n) = n(1^n - \log n + \sqrt{n^2}) \quad \text{for all } n > 0$$

(c)

$$f(n) = \begin{cases} 2^n & 0 < n \leq 3 \\ 5n! + 8 & 3 < n \leq 52 \\ 4092 + n & n > 52 \end{cases}$$

(d)

$$f(n) = \begin{cases} 3n + 2^n + 4 & 0 < n \leq 4 \\ \sqrt{n} + n^2\sqrt{n} & 4 < n \leq 16 \\ 1028 & n > 16 \end{cases}$$



2. (10 points) Give three disadvantages of using benchmarking to determine algorithmic time complexity.

3. (10 points) Why must the `Chain<T>` member functions be in the same file as the class definition?

4. Consider the following algorithm:

```
2 // Calculates the sum of all the values from 1 up to
  // (and including) n
unsigned int summation( unsigned int n )
4 {
  unsigned int ans = n;
6   for(int i = n; i<n+1; ) {
      ans *= ++i;
8   }
  return ans/2;
10 }
```

(a) (5 points) Give the big-oh notation for the time complexity of the above algorithm. Explain your reasoning.

(b) (10 points) Write a recursive function that returns the sum of all the integers from 1 up to (and including) n .

(c) (5 points) Using big-oh notation, describe the the worst case time complexity for the above algorithm. Explain your reasoning.



5. In the first laboratory assignment, you created a simple spell checking program.
(a) (10 points) Describe how your program would handle the following “words”:
3.1459 and **35min**.

(b) (10 points) How could your program be improved to handle these possibilities
in a more intelligent way?

6. (20 points) Using big-oh notation, describe the the worst case time complexity for the algorithm below. Explain your reasoning.

```
void coolSort( vector<unsigned int>& data )
2 {
  bool flag = true;
4  int divisor = 1;
  while( flag ) {
6    vector< list<unsigned int> > buckets(10);
    flag = false;
8    vector<unsigned int>::iterator start=data.begin();
    vector<unsigned int>::iterator stop=data.end();
10   while( start!=stop ) {
    int index = (* start/divisor)%10;
12    if( (* start/divisor)>0 ) {
    flag = true;
14    }
    buckets[index].push_back(* start);
16    ++start;
  }
18  start=data.begin();
  for(int i=0; i<10; ++i) {
20    list<unsigned int>::iterator lstart = buckets[i].begin();
    list<unsigned int>::iterator lstop = buckets[i].end();
22    while( lstart!=lstop ) {
    *start = *lstart;
24    ++start;
    ++lstart;
26    }
  }
28  divisor += 10;
30 }
```



bonus. (10 points) Show that the algorithm in the previous problem will sort a vector with the following integers: 216, 521, 425, 116, and 24.



Additional work area for any problem. Clearly identify to which problem the work on this page is related.



Additional work area for any problem. Clearly identify to which problem the work on this page is related.