

University of Nevada, Las Vegas Computer Science 477/677 Fall 2022

Assignment 4: Due Friday September 14, 2022, Midnight

This is the final version.

Name: _____

You are permitted to work in groups, get help from others, read books, and use the internet. Please follow Mr. Wang's instructions on how to submit your completed assignment.

1. Fill in the blanks. One word per blank.

(a) The two standard operators of the ADT *stack* are _____ and _____.

(b) The two standard operators of the ADT *array* are _____ and _____.

2. Find the asymptotic time complexity of each code fragment. Give the answer using Θ notation.

(a) `for(int i = 0; i*i < n; i++)`

(b) `for(float x = n; x > 2.0; x = sqrt(x))`

(c) `for(int i = 1; i < n; i = 2*i)
 for(int j = 2; j < i; j = j*j);`

(Hint: use substitution)

3. Solve the recurrences. Give asymptotic answers in terms of n .

(a) $F(n) = F(n/2) + 1$

(b) $F(n) = 2F(n/2) + 1$

(c) $F(n) = 2F(n/2) + n$

(d) $F(n) = n + F(n/5) + F(7n/10)$

(e) $F(n) = F(3n/4) + F(n/2) + 3F(n/4) + n$

(f) $F(n) = F(n - 1) + \sqrt{n}$

(g) $F(n) = F(n - \log n) + \log n$

(h) $F(n) = F(\log n) + 1$

(i) $F(n) = 2F(n/2) + 2F(n/6) + F(2n/3) + n^2$

4. Consider the function $f(n)$ computed by the code below.

```
int f(int n)
{
    // input condition: n >= 0
    if(n==0) return 1;
    else return f(n/2)+f(n/4)+f(n/4+1)+n;
}
```

The function can be computed by recursion, as given in the C++ code above. However, we could also compute $f(n)$ using dynamic programming or memoization.

(a) What is the asymptotic value of $f(n)$? The value itself, not the time to compute it. Write the recurrence, then solve it.

(b) What is the asymptotic time complexity of the recursive computation of $f(n)$, using the code given? Write the recurrence, then solve it.

(c) What is the asymptotic time complexity of the dynamic programming computation of $f(n)$?

(d) What is the asymptotic time complexity of the computation of $f(n)$ using memoization?

5. You are given an array A of N integers, where N could be any positive integer. Write C++ code that prints out the sum of the entries of A, then write C++ code that prints out the maximum entry of A. The code must be correct C++ code.

```
int const N = 1000; // or whatever
int A[N];

int main()
{
    cout << "The sum of the array A is " << sumA;
    cout << "The maximum of the array A is " << maxA;
    return 1;
}
```

6. You are given a sequence X of N positive numbers. Give an algorithm which finds the maximum total subsequence, subject to the condition that the subsequence contains no two consecutive terms of the original sequence. Write your algorithm in correct C++ code.

```
int const N = 1000; // or whatever
int X[N]; // input sequence (global)
int back[N]; // back pointers (global)
// additional array or arrays or variables as needed

void printsubsequenc(What parameters do you need?)
{
    // uses backpointers to print the output subsequence in forward order.
}
```