**Due Date**: Wednesday October 21, 2020, 11:59pm

## Instructions

Answer all questions on paper or a tablet using your own handwriting. Put your name, student ID number and page number at the top of each page. If you use paper make a photo of each page and upload your solutions to crowdmark. If you use a tablet, export your assignment to .pdf and upload the .pdf to crowdmark.

### Textbook Reading

- Sections: 10.2, 10.3
- DivAndConq.pdf notes (under Lecture Notes on Canvas)

### Definitions, Concepts & Keywords

- Solve first order non-homogenous recurrences.
- Solve some second order non-homogenous recurrences.
- Understand the Mergesort algorithm.
- Solve recurrences arising from divide-and-conquer algorithms.

### Exercises

A. **Textbook Questions**

Section 10.2 Exercises 1(a+b+d), 4, 24.
Section 10.3 Exercises 1(b+d), 8.

B. **Questions**

1. Second order homogenous recurrence question.
2. John takes out a bank loan for \$20,000 to buy a car. If the bank charges interest at 10% per year, and John has to pay back the loan over 5 years, how much does he have to pay every year? Use a recurrence. See example 10.29.
3. Consider the complete bipartite graph $K_{n,n}$ with $2n$ vertices. Let $k_n$ be the number of edges in $K_{n,n}$. Draw $K_{1,1}$, $K_{2,2}$ and $K_{3,3}$ and determine $k_1$, $k_2$, $k_3$. Give a recurrence relation for $k_n$ and solve it using an initial value.
4. Calculate a formula for $\sum_{i=0}^{n-1} 3i^2$ and $\sum_{k=0}^{n-1} k2^k$.
5. Solve the recurrence $a_n = 2a_{n-1} + 4^{n-1}$ with $a_1 = 3$.
6. Below is C code and Python code for an algorithm.

C code:

```
1: long crazy(long A[],long n) {
2:     long i,s,t;
3:     if( n<=1 ) return A[0];
4:     s = crazy(A,n-1);
5:     for( i=1; i<n; i++ )
6:         A[i-1] += A[i];
7:     t = crazy(A,n-1);
8:     return s+t;
9: }
```

Python code:

```
1: def crazy(A,n):
2:       if n<=1:
3:           return A[0]
4:       s = crazy(A,n-1)
5:       for i in range(1,n):
6:           A[i-1] += A[i]
7:       t = crazy(A,n-1)
8:       return s+t
```

Let $c_n$ be the number of times line 6 is executed. Assume $n = 2^k$ and $k \geq 0$. Give a recurrence equation and an initial value for $c_n$ and solve it. Check that $c_4 = 11$. Note, you do not need to know what the code is doing to answer this question.

7. Consider the recurrence $a_n - 3a_{n-1} + 2a_{n-2} = n$.
   (a) Find the general solution to the associated homogenous recurrence.
   (b) Find a particular solution.
   (c) Find the solution for $a_0 = a_1 = 1$.
8. Consider the recurrence $a_n - a_{n-1} - 6a_{n-2} = 5 \cdot 3^n$.
   (a) Find the general solution to the associated homogenous recurrence.
   (b) Find a particular solution.
   (c) Find the solution for $a_0 = 7$ and $a_1 = 5$.
9. The recurrence for number of comparisons that the Mergesort algorithm does, assuming $n = 2^k$, is $C(n) \leq 2\,C(n/2) + n - 1$ and $C(1) = 0$. If the input array $A$ is already sorted, give a recurrence for the number of comparisons Mergesort does and solve it. Hint: how many comparisons does the merging algorithm do?
10. Below is C code and Python code for an algorithm.

    C code

```
1: int locate(int A[],int s,int n,int x)
2: {  if( n==1 )
3:        if( A[s]==x ) return s;
4:        else return -1;
5:     int j,k,m = n/2;
6:     j = locate(A,s,m,x);
7:     k = locate(A,s+m,n-m,x);
8:     return j>=0 ? j : k;
9: }
```

    Python code

```
1: def locate(A,s,n,x):
2:     if n==1:
3:         if A[s]==x: return s
4:         else: return -1
5:     m = n/2;
6:     j = locate(A,s,m,x);
7:     k = locate(A,s+m,n-m,x);
8:     if j>=0: return j
9:     else: return k
```

    Let $a_n$ be the number of times the comparison `A[s]==x` is executed in line 3. For $n = 2^k$ with $k \geq 0$ give a recurrence equation and initial value for $a_n$ and solve it. Note, you do not need to know what the code is doing to answer this question.
11. For $n = 2^k$ the recurrence for a divide-and-conquer algorithm is

$$T_n = 4T_{n/2} + 2n \;\text{ for } n > 1.$$

    Solve it for the initial value $T_1 = 1$. Check that $T_4 = 40$.

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} \qquad \sum_{k=1}^{n} k\binom{n}{k} = \frac{n}{2}2^n \qquad \sum_{k=1}^{n} k2^{k-1} = (n-1)2^n + 1.$$

Table 1: Some useful sums